L Number	Hits	Search Text	DB	Time stamp
1	8	((multicast$4 multi-cast$4) and unicast$4 and (link adj state) and ospf and mospf).detd.	USPAT; EPO; JPO; DERWENT; IBM_TDB	2003/06/02 08:31
2	0	((multicast$4 multi-cast$4) and unicast$4 and (link adj state) and (ospf or (short$4 adj path)) and mospf) and router	EPO; JPO; DERWENT; IBM_TDB	2003/06/02 08:32
3	0	((multicast$4 multi-cast$4) and unicast$4 and (ospf or (short$4 adj path)) and mospf) and router	EPO; JPO; DERWENT; IBM_TDB	2003/06/02 08:39
4	0	unicast$4 same (ospf or (short$4 adj path) or mospf) same (multicast adj router)	EPO; JPO; DERWENT; IBM_TDB	2003/06/02 08:55
5	1	unicast$4 same (multicast adj router)	EPO; JPO; DERWENT; IBM_TDB	2003/06/02 08:56
6	18	unicast$4 same (multicast adj router)	USPAT	2003/06/02 09:38
7	18	(unicast$4 and (multicast adj router) and (table database (data adj base))) and (unicast$4 same (multicast adj router))	USPAT	2003/06/02 10:01
8	21	((multicast adj router) same (table database (data adj base)))	USPAT	2003/06/02 10:07
9	4	(multicast adj router) same (table database (data adj base)) same ((short$4 adj path) or ospf or mospf)	USPAT	2003/06/02 10:12
10	0	(multicast adj router) same (table database (data adj base)) same ((short$4 adj path) or ospf or mospf)	EPO; JPO; DERWENT; IBM_TDB	2003/06/02 10:13
11	0	(multicast adj router) same (table database (data adj base)) same ((short$4 adj (path route link)) or ospf or mospf)	EPO; JPO; DERWENT; IBM_TDB	2003/06/02 10:13
12	4	(multicast adj router) same (table database (data adj base)) same ((short$4 adj (path route link)) or ospf or mospf)	USPAT	2003/06/02 10:14
13	5	(mrouter ((multicast multi-cast) adj router)) same (table database (data adj base)) same ((short$4 adj (path route link)) or ospf or mospf)	USPAT	2003/06/02 10:43
14	5	(mrouter ((multicast multi-cast) adj router)) same (table database (data adj base)) same (short-path (short$4 adj (path route link)) or ospf or mospf)	USPAT	2003/06/02 10:47
15	120	(multicast mrouter ((multicast multi-cast) adj router)) same (table database (data adj base)) same (short-path (short$4 adj (path route link)) or ospf or mospf or router)	USPAT	2003/06/02 10:48
16	13	(multicast mrouter ((multicast multi-cast) adj router)) same (table database (data adj base)) same (short-path (short$4 adj (path route link)) or ospf or mospf) same router	USPAT	2003/06/02 11:31

log search

```
      (FILE 'HOME' ENTERED AT 11:02:07 ON 02 JUN 2003)

      FILE 'PCTFULL' ENTERED AT 11:02:15 ON 02 JUN 2003
L1             1 S SPT(P)(TABLE OR DATABASE)(P)((MULTICAST(A)ROUTER) OR MROUTER OR MSROUTER)

      FILE 'EUROPATFULL' ENTERED AT 11:04:06 ON 02 JUN 2003
L2             0 S SPT(P)(TABLE OR DATABASE)(P)((MULTICAST(A)ROUTER) OR MROUTER OR MSROUTER)
L3             0 S SPT(P)(TABLE OR DATABASE)(P)(((MULTICAST MULTI-CAST)(A)ROUTER) OR MROUTER OR MSROUTER)
L4             1 S (SPT OR SHORT?(A)PATH)(P)(TABLE OR DATABASE)(P)(((MULTICAST MULTI-CAST)(A)ROUTER) OR MROUTER OR
MSROUTER)

      FILE 'INSPEC' ENTERED AT 11:08:02 ON 02 JUN 2003
L5             0 S (SPT OR SHORT?(A)PATH)(P)(TABLE OR DATABASE)(P)(((MULTICAST MULTI-CAST)(A)ROUTER) OR MROUTER OR
ROUTER)

      FILE 'COMPENDEX' ENTERED AT 11:08:23 ON 02 JUN 2003
L6             0 S (SPT OR SHORT?(A)PATH)(P)(TABLE OR DATABASE)(P)(((MULTICAST MULTI-CAST)(A)ROUTER) OR MROUTER OR
MSROUTER)
L7             0 S (SPT OR SHORT?(A)PATH)(P)STORING(P)(((MULTICAST MULTI-CAST)(A)ROUTER) OR MROUTER OR MSROUTER)

      FILE 'EUROPATFULL' ENTERED AT 11:09:31 ON 02 JUN 2003
L8             0 S (SPT OR SHORT?(A)PATH)(P)STORING(P)(((MULTICAST MULTI-CAST)(A)ROUTER) OR MROUTER OR MSROUTER)
               SET LINELENGTH 250
```

Citation

ACM SIGCOMM Computer Communication Review >archive
Volume 25 , Issue 1 (January 1995) >toc
Special twenty-fifth anniversary issue. Highlights from 25 years of the Computer Communication Review

Multicast routing in internetworks and extended LANs

Author
Stephen E. Deering

Publisher
ACM Press New York, NY, USA

Pages: 88 - 101 Periodical-Issue-Article
Year of Publication: 1995
ISSN:0146-4833

http://doi.acm.org/10.1145/205447.205457 (Use this link to Bookmark this page)

↑ **ABSTRACT**

Multicasting is used within local-area networks to make distributed applications more robust and more efficient. The growing need to distribute applications across multiple, interconnected networks, and the increasing availability of high-performance, high-capacity switching nodes and networks, lead us to consider providing LAN-style multicasting across an internetwork. In this paper, we propose extensions to two common internetwork routing algorithms---distance-vector routing and link-state routing---to support low-delay datagram multicasting. We also suggest modifications to the single-spanning-tree routing algorithm, commonly used by link-layer bridges, to reduce the costs of multicasting in large extended LANs. Finally, we show how different link-layer and network-layer multicast routing algorithms can be combined hierarchically to support multicasting across large, heterogeneous internetworks.

↑ **INDEX TERMS**

Primary Classification:
C. Computer Systems Organization

↳ C.2 COMPUTER-COMMUNICATION NETWORKS

Additional Classification:
C. Computer Systems Organization
↳ C.1 PROCESSOR ARCHITECTURES
↳ C.1.2 Multiple Data Stream Architectures (Multiprocessors)
↳ Subjects: Interconnection architectures (e.g., common bus, multiport memory, crossbar switch)
↳ C.2 COMPUTER-COMMUNICATION NETWORKS
↳ C.2.0 General
↳ Subjects: Data communications

General Terms:
Algorithms, Design, Performance

↑ **Peer to Peer - Readers of this Article have also read:**

Editorial pointers
Communications of the ACM 44, 9
Diane Crawford

News track
Communications of the ACM 44, 9
Robert Fox

Forum
Communications of the ACM 44, 9
Diane Crawford

New Products
Linux Journal 1996, 27es
CORPORATE Linux Journal Staff

Editorial
interactions 8, 5
Steven Pemberton

Multicast Routing in Internetworks and Extended LANs

S. Deering

(Originally Published in: Proc. SIGCOMM '88, Vol 18, No. 4, August 1988)

Multicast Routing in Internetworks and Extended LANs

Stephen E. Deering
Computer Systems Laboratory
Stanford University

Abstract

Multicasting is used within local-area networks to make distributed applications more robust and more efficient. The growing need to distribute applications across multiple, interconnected networks, and the increasing availability of high-performance, high-capacity switching nodes and networks, lead us to consider providing LAN-style multicasting across an internetwork. In this paper, we propose extensions to two common internetwork routing algorithms—distance-vector routing and link-state routing—to support low-delay datagram multicasting. We also suggest modifications to the single-spanning-tree routing algorithm, commonly used by link-layer bridges, to reduce the costs of multicasting in large extended LANs. Finally, we show how different link-layer and network-layer multicast routing algorithms can be combined hierarchically to support multicasting across large, heterogeneous internetworks.

1 Introduction

The multicast capability of local-area networks such as Ethernet [8] provides two important benefits to distributed applications:

1. When an application must send the same information to more than one destination, multicasting is more efficient than unicasting: it reduces the transmission overhead on the sender and the network, and it reduces the time it takes for all destinations to receive the information.

2. When an application must locate, query, or send information to one or more hosts whose addresses are unknown or changeable, multicasting serves as a simple, robust alternative to configuration files, name servers, or other binding mechanisms.

Multicasting applications have proliferated in those environments in which the multicast capability has been made available to application programmers, whether in the form of process groups in the V System [5], UDP broadcast sockets in Berkeley UNIX [20], or NetBIOS multicast datagrams in MS-DOS [16]. In some cases, multicasting has played an important role in organizing the underlying operating systems and protocols themselves, as well as being offered as a service for applications.[1]

For networks in which all hosts share a common transmission channel, such as bus, ring, or satellite networks, the multicast capability is provided trivially and at the same cost to the network as unicasting. When such networks are interconnected by store-and-forward packet switches, multicasting across the resulting internetwork often requires the commitment of additional switching and transmission resources, beyond those required for unicasting. However, as those resources become more abundant, in the form of fast packet switches, cheap memories, and high-bandwidth local and long-haul communication links, an economic argument for denying users the benefits of an internetwork multicast capability becomes harder to sustain.

Link-layer bridges, such as the DEC LANBridge 100 [12] and the Vitalink TransLAN [11], have taken advantage of the improving economics of communication to extend LAN performance and LAN functionality—including multicast—across multiple networks. That is not yet the case with network-layer routers, such as DoD IP Gateways [14] or ISO Intermediate Systems [18]. Therefore, when moving multicast-based applications to an environment that includes network-layer routers, it is currently necessary to give up the efficiency of multicasting and to replace the flexible binding capability of multicasting with more complicated or fragile mechanisms. This paper addresses that problem by proposing extensions to two common routing algorithms used by network-layer routers—distance-vector routing and link-state routing—to provide LAN-style multicasting across datagram-based internetworks. We also suggest modifications to link-layer bridge routing to improve the effi-

[1] Some of these systems have implemented multicasting by using the local-area network's *broadcast* facility, relying on software filtering in the receiving hosts. This approach incurs undesirable overhead on those hosts that must receive and discard unwanted packets, overhead that gets worse as more and more applications use multicasting. Fortunately, this problem can be avoided in modern LANs, such as Ethernet and other networks conforming to the IEEE 802 [15] standards, which provide multicast addresses that can be recognized and filtered by host interface hardware.

ciency of multicasting in large extended LANs.

In the next section of this paper we define what we mean by "LAN-style multicasting." In Section 3 we describe the environment in which multicast routing is to take place. Then follow three sections, describing specific multicast extensions to the single-spanning-tree, distance-vector, and link-state routing algorithms. In Section 7, we describe how a variety of link-layer and network-layer multicast routing schemes may be combined to support multicasting in a large, heterogeneous internetwork. In Section 8 we call attention to other work in the same area, and in the concluding section we summarize our results and point the way to further work.

2 Desired Properties for Internetwork Multicasting

Existing multicast-based distributed applications have been developed in the LAN environment. To support the migration of such applications to an internetwork environment, it is desirable to retain, to the degree possible, the following important properties of LAN multicasting:

- *Group addressing.* In a LAN, a multicast packet is sent to a group address which identifies a set of destination hosts. The sender need not know the membership of the group and need not itself be a member of the group. There is no restriction on the number or location of hosts in a group. Hosts can join and leave groups at will, with no need to synchronize or negotiate with other members of the group or with potential senders to the group.

  With such group addressing, multicasting can be used for such purposes as locating a resource or a server when its specific address is unknown, searching for information among a dynamically-changing set of information providers, or distributing information to an arbitrarily-large, self-selected set of information consumers.

- *High probability of delivery.* In a LAN, the probability that a member of a group successfully receives a multicast packet sent to the group is usually the same as the probability that the member successfully receives a unicast packet sent to its individual address. Furthermore, that probability of successful reception by every member is very high, in the absence of partitioning. This property allows the designers of end-to-end reliable multicast protocols to assume that a small number of retransmissions of a multicast packet will result in successful delivery to all destination group members that are up and reachable. The probability of damage, duplication, or misordering of multicast packets in a LAN is very low, but not necessarily zero; recovery from such events is also the responsibility of end-to-end protocols, to the extent required by particular applications.

  The probability of successful multicast delivery in an internetwork may well decrease as the distance between sender and group members increases, but it must stay within bounds that allow successful recovery by end-to-end protocols.

- *Low delay.* LANs impose very little delay on the delivery of multicast packets. This is an important property for a number of multicast applications, such as distributed conferencing, parallel computing, and resource location. Also, the delay between when a host decides to join a group and when it can start receiving packets addressed to that group, called the *join latency*, is very low in a LAN, usually just the time required to update a local address filter. Low join latency is important for certain applications, such as those that use multicasting to communicate with migrating processes or mobile hosts.

  The delay properties of large internetworks are, inevitably, worse than LANs because of their greater geographic extent and their greater number of links and switches. However, the use of high-speed packet switches and low-delay long distance communication links such as optical fibers has the potential to significantly reduce the gap between local-area network and internetwork delay characteristics. In order to exploit that potential, it is important that internetwork multicast routing algorithms produce low-delay routes, in preference to routes that maximize bandwidth or minimize network resource consumption. The availability of bandwidth and other network resources keeps improving; delay is the limiting factor for wide-area communication.

The large scale and multi-hop nature of internetworks motivates a simple extension to LAN multicasting semantics to allow senders to limit the distance a multicast packet may travel. Internetwork datagram protocols, such as DoD IP [24] and ISO CLNP [17], include a *time-to-live* (TTL) field in the packet header for the purpose of bounding the amount of time a packet may be in transit. By using a very small TTL value, a sender may limit the "scope" of a multicast packet to reach nearby group members only.[2] This can be of benefit to the internetwork, by reducing the amount of multicast traffic that has

[2] An interesting and useful application of TTL scope control is "expanding ring searching", a concept described by Boggs in his dissertation on internetwork broadcasting [3]. An example of its use is searching for the nearest name server: a host multicasts a name server query, starting with a TTL that reaches only its immediate neighborhood, and incrementing the TTL on each retransmission to reach further and further afield, until it receives a reply.

to be carried long distances, and it can be of benefit to the sender, by reducing the number of responders when querying a large group. Even when it is desired to reach an entire group, if the sender knows that all the members are nearby, use of a small TTL can help to reduce the delivery costs incurred under some multicast routing schemes.

3 Assumed Environment for Internetwork Multicasting

We assume an environment of multi-access networks (LANs and, possibly, satellite networks) interconnected in an arbitrary topology by packet switching nodes (bridges and/or routers). Point-to-point links (both physical links such as fiber-optic circuits and virtual links such as X.25 virtual circuits) may provide additional connections between the switching nodes, or from switching nodes to isolated hosts, but almost all hosts are directly connected to LANs.

The LANs are assumed to support *intra*network multicasting. The hosts have address filters in their LAN interfaces which can recognize and discard packets destined to groups in which the hosts have no interest, without interrupting host processing. Bridges and routers attached to LANs are capable of receiving *all* multicast packets carried by the LAN, regardless of destination address.

Link-layer bridges perform their routing function based on LAN addresses that are unique across the collection of interconnected LANs. Network-layer routers perform routing based on globally-unique internetwork addresses which are mapped to locally-unique LAN addresses for transmission across particular LANs. We assume that globally-unique internetwork *multicast* addresses can be mapped to corresponding LAN multicast addresses according to LAN-specific mapping algorithms. Ideally, each internetwork multicast address maps to a different LAN address; in cases where address-space constraints on a particular LAN force a many-to-one mapping of internetwork to LAN multicast addresses, the hosts' address filters may be less effective, and additional filtering must be provided in host software.

4 Single-Spanning-Tree Multicast Routing

Link-layer bridges [11, 12] transparently extend LAN functionality across multiple interconnected LANs, possibly separated by long distances. To maintain transparency, bridges normally propagate every multicast and broadcast packet across every segment of the extended LAN. This is considered by some to be a *dis*advantage of bridges, because it exposes the hosts on each segment to the total broadcast and multicast traffic of all the segments. However, it is the misguided use of broadcast packets, rather than multicast packets, that is the threat to host resources; multicast packets can be filtered out by host interface hardware. Therefore, the solution to the host exposure problem is to convert broadcasting applications into multicasting applications, each using a different multicast address.

Once applications have been converted to use multicast, it is possible to consider conserving bridge and link resources by conveying multicast packets across only those links necessary to reach their target membership. In small bridged LANs, bridge and link resources are usually abundant; however, in large extended LANs that include lower-bandwidth long-haul links or that have a lot of multicast traffic for groups that reside in small subregions of the extended LAN, it may be of great benefit not to send multicast packets everywhere.

Bridges typically restrict all packet traffic to a single spanning tree, either by forbidding loops in the physical topology or by running a distributed algorithm among the bridges to compute a spanning tree [23]. When a bridge receives a multicast or broadcast packet, it simply forwards it onto every incident branch of the tree except the one on which it arrived. Because the tree spans all segments and has no loops, the packet is delivered exactly once (in the absence of errors) to every segment.

If bridges knew which of their incident branches led to members of a given multicast group, they could forward packets destined to that group out those branches only. Bridges are able to learn which branches lead to *individual* hosts by observing the source addresses of incoming packets. If group members were to periodically issue packets with their group address as the source, the bridges could apply the same learning algorithm to group addresses.

For example, assume that there is an *all-bridges* group B to which all bridges belong. Each host that is a member of a group G may then inform the bridges of its membership by periodically transmitting a packet with source address G, destination address B, packet type *membership report*, and no user data.

Figure 1 shows how this works in a simple bridged LAN with a single group member. LANs a, b, and c are bridged to a backbone LAN d. Any membership report issued by the one group member on LAN a is forwarded to the backbone LAN by the bridge attached to a, to reach the rest of the *all-bridges* group. There is no need to forward the membership report to LANs b or c because they are leaves of the spanning tree which do not reach any additional bridges. (Bridges are able to identify leaf LANs either as a result of their tree-building algorithm or

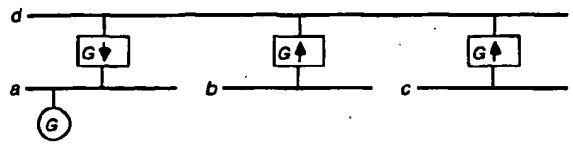


Figure 1: Bridged LAN with One Group Member

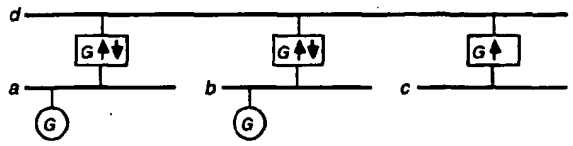


Figure 2: Bridged LAN with Two Group Members

by periodically issuing reports of their own membership in the *all-bridges* group.)

After the membership report has reached all bridges, they each know which direction leads to the member of G, as illustrated by the arrows in Figure 1. Subsequent transmission of multicast packets destined to G are forwarded only in the direction of that membership. For example, a multicast packet to G originating on LAN b will traverse d and a, but not c. A multicast to G originating on a will not be forwarded at all.

Figure 2 shows the state of bridge knowledge after a second member joins the group on LAN b. Now multicast packets to G will be conveyed towards LANs a and b, but not towards c.

This multicast routing algorithm requires little extra work or extra space in the bridges. Typical learning bridges maintain a table of *unicast* addresses. Each table entry is a triple:

(*address, outgoing-branch, age*)

where the *age* field is used to detect stale data. The source address and source branch of each incoming packet is installed in the table, and the destination address of each arriving unicast packet is looked up in the table to determine an outgoing branch. To support multicasting, the table must also hold multicast addresses. As seen in Figure 2, a single multicast address may have multiple outgoing branches (and *age* fields, as discussed below), so the table entries become variable-length records of the form:[3]

(*address*, (*outgoing-branch, age*),
(*outgoing-branch, age*), ...)

[3]Many bridges are designed to connect only two, or some other small number, of links; for them, it may be acceptable to use fixed, maximum-sized records, in order to simplify memory management.

An arriving group membership report causes a table entry for its source address to be installed or updated. The destination address of an arriving multicast packet is looked up in the table to determine the set of outgoing branches. The branch over which the multicast packet arrived is always deleted from the set of outgoing branches before forwarding.

The *age* field in table entries for multicast addresses is handled somewhat differently than for unicast addresses. When a bridge receives a unicast packet, if its destination address is absent from the table, or if its table entry has expired (i.e., its *age* exceeds some threshold), the packet is forwarded out *all* branches except the incoming one. It is expected that responding traffic from the destination will later allow the bridge to learn its location. When a bridge receives a multicast packet, on the other hand, it forwards the packet over only those branches that are identified by non-expired table entries. Expired entries are treated as evidence that there are no longer any members reachable over that branch. Therefore, group members must regularly report their memberships at intervals less than the membership expiry threshold.

The overhead of membership reporting traffic is determined by the choice of reporting interval T_{report}—the larger T_{report}, the less the reporting overhead. On the other hand, choosing a large T_{report} has the following drawbacks:

- The expiry threshold T_{expire} for bridge table entries should be a multiple of T_{report} in order to tolerate occasional loss of membership reports. The larger T_{expire}, the longer a bridge will continue to forward multicast packets onto a particular branch after there are no longer any members reachable along that branch. This is not particularly serious, given that hosts are protected from unwanted traffic by their address filters.
- If a host is the first member of a group on a particular LAN and its first one or two membership reports are lost due to transmission errors, the bridges will be unaware of its membership until one or two times T_{report} has passed. This fails to meet the goal of low join latency, stated in Section 2. It can be avoided by having hosts issue several membership reports in close succession when they first join a group.
- If the spanning tree changes due to a bridge or LAN coming up or going down, the multicast entries in the bridge tables may become invalid for as long as T_{expire}. This problem can be avoided by having the bridges revert to broadcast-style forwarding for a period of T_{expire} after any topology change.

Therefore, none of these drawbacks is serious enough to

prevent the use of a relatively large T_{report}, say on the order of minutes rather than seconds.

There is another technique that can be used to reduce the reporting traffic, apart from increasing T_{report}. When issuing a membership report for group G, a host initializes the destination address field to G, rather than the *all-bridges* address. The bridge(s) directly attached to the reporting member's LAN then replace the G with the *all-bridges* address before forwarding to the other bridges. (A bridge can recognize such reports by the fact that the source and destination are the same group address.) This allows other members of the same group on the same LAN to overhear the membership report and suppress their own, superfluous reports. In order to avoid unwanted synchronization of membership reports, whenever such a report is transmitted on a LAN all members of the reported group on that LAN set their next report timer to a random value in a range around T_{report}. The next report for that group is issued by whichever member times out first, at which time new random timeouts are again chosen. Thus, the reporting traffic originating on each LAN is reduced to one report per group present, rather than one report from *every member* of every group present, in every T_{report} period. This is a significant reduction in the common case where a single group has more than one member on a single LAN.

To get a feeling for the costs of this algorithm, assume that a typical extended LAN consists of 10 segments, on which each host belongs to 5 groups, each segment has members of 20 different groups, there are 50 groups in total, and the membership reporting interval T_{report} is 200 seconds. Then:

- The overhead on hosts is the transmission or reception of one membership report packet every 40 seconds.
- The overhead on leaf segments and on bridge interfaces to leaf segments is one membership report packet every 10 seconds.
- The overhead on non-leaf segments and on bridge interfaces to non-leaf segments is the sum of the reporting traffic from each segment, that is one membership report packet every second.
- The storage overhead in each bridge is 50 group address entries.

Such costs are insignificant compared to the available bandwidth and bridge capacity in current extended LAN installations. Furthermore, the overheads on hosts and leaf segments are independent of the total number of segments; extended LANs with hundreds of segments would see greater overheads only on the "backbone" segments, not on the (presumably) more numerous leaf segments to which most hosts would be connected.

The bridge multicast routing algorithm as described requires that hosts be modified to issue membership reports for those groups they belong to. This compromises the transparency property that is one of the important features of link-layer bridges. However, if hosts are to be modified anyway to use multicast rather than broadcast, the membership reporting protocol might reasonably be implemented at the same time. The reporting is best handled at the lowest level in the host operating system, such as the LAN device driver, in order to minimize host overhead. Future LAN interfaces might well provide the membership reporting service automatically, without host involvement, as a side-effect of setting the multicast address filter. Conversely, non-conforming hosts might be accommodated by allowing manual insertion of membership information into individual bridge tables.

5 Distance-Vector Multicast Routing

The distance-vector routing algorithm, also known as the Ford-Fulkerson [9] or Bellman-Ford [2] algorithm, has been used for many years in many networks and internetworks. For example, the original Arpanet routing protocol [22] was based on distance-vector routing, as was the Xerox PUP Internet [4] routing protocol. It is currently in use by Xerox Network Systems internetwork routers [27], some DARPA Internet core gateways [14], and numerous UNIX systems running Berkeley's *routed* internetwork routing process [13], to name only a few.

Routers that use the distance-vector algorithm maintain a routing table, which contains an entry for every reachable destination in the internetwork. A "destination" may be a single host, a single subnetwork, or a cluster of subnetworks. A routing table entry typically looks like:

(*destination, distance, next-hop-address, next-hop-link, age*)

Distance is the distance to the destination, typically measured in hops or some other unit of delay. *Next-hop-address* is the address of the next router on the path towards the destination, or the address of the destination itself if it shares a link with this router. *Next-hop-link* is a local identifier of the link used to reach *next-hop-address*. *Age* is the age of the entry, used to time out destinations that become unreachable.

Periodically, every router sends a routing packet out each of its incident links. For LAN links, the routing packet is usually sent as a local broadcast or multicast in order to reach all neighboring routers. The packet contains a list of (*destination, distance*) pairs (a "distance vector") taken from the sender's routing table. On receiving a routing packet from a neighboring router, the

receiving router may update its own table if the neighbor offers a new, shorter route to a given destination, or if the neighbor no longer offers a route that the receiving router had been using. By this interaction, routers are able to compute shortest-path routes to all internetwork destinations. (This brief description leaves out several details of the distance-vector routing algorithm which are important, but not relevant to this presentation. Further information can be found in the references cited above.)

One straightforward way to support multicast routing in a distance-vector routing environment would be to compute a single spanning-tree across all of the links and then use the multicast routing algorithm described in the previous section. The spanning tree could be computed using the same algorithm as link layer bridges or, perhaps, using one of Wall's algorithms [26] for building a single tree with low average delay. However, in a general topology that provides alternate paths, no single spanning tree will provide minimum-delay routes from all senders to all sets of receivers. In order to meet our goal of low-delay multicasting, and to provide reasonable semantics for TTL scope control, we require that a multicast packet be delivered along a shortest-path (or an almost-shortest-path) tree from the sender to the members of the multicast group.

There is potentially a different shortest-path tree from every sender to every multicast group. However, every shortest-path multicast tree rooted at a given sender is a subtree of a single shortest-path *broadcast* tree rooted at that sender. In this section, we use that observation as the basis for a number of refinements to Dalal and Metcalfe's *reverse-path-forwarding* broadcast algorithm [6] which take advantage of the distance-vector routing environment to provide low-delay, low-overhead multicast routing.

5.1 Reverse Path Flooding (RPF)

In the basic reverse path forwarding algorithm, a router forwards a broadcast packet originating at source S if and only if it arrives via the shortest path from the router back to S (i.e., the "reverse path"). The router forwards the packet out all incident links except the one on which the packet arrived. In networks where the "length" of each path is the same in both directions, for example when using hop counts to measure path length, this algorithm results in a shortest-path broadcast to all links.

To implement the basic reverse path forwarding algorithm, a router must be able to identify the shortest path from the router back to any host. In internetworks that use distance-vector routing for unicast traffic, that information is precisely what is stored in the routing tables in every router. Furthermore, most implementations of distance-vector routing use hop counts as their distance measure. Thus, reverse path forwarding is easily implemented and effective at providing shortest-path broadcasting in most distance-vector routing environments. (Distance metrics other than hop counts may also support shortest-path or almost-shortest-path broadcasting, as long as the resulting path lengths are the same or almost the same in both directions.)

As described, reverse path forwarding accomplishes a *broadcast*. To use the algorithm for multicasting, it is enough simply to specify a set of internetwork multicast addresses, that can be used as packet destinations, and perform reverse path forwarding on all packets destined to such addresses. Hosts choose which groups they wish to belong to, and simply discard all arriving packets addressed to any other group.

The reverse path forwarding algorithm as originally specified in [6] assumes an environment of point-to-point links between routers, with each host attached to its own router. In the internetwork environment of interest here, routers may be joined by multi-access links as well point-to-point links, and the majority of hosts reside on multi-access links (LANs). It is possible and desirable to exploit the multicast capability of those multi-access links to reduce delay and network overhead, and to allow host interface hardware to filter out unwanted packets. To accomplish this, whenever a router (or an originating host) forwards a multicast packet onto a multi-access link, it sends it as a local multicast, using an address derived from the internetwork multicast destination address. In this way, a single packet transmission can reach all member hosts that may be present on the link. Routers are assumed to be able to hear all multicasts on their incident links, so the single transmission also reaches any other routers on that link. Following the reverse path algorithm, a receiving router forwards the packet further only if it considers the sending router to be on the shortest path, i.e., if the sending router is the *next-hop-address* to the originator of the multicast.

The major drawback of the basic reverse path forwarding algorithm (as a broadcast mechanism) is that any single broadcast packet may be transmitted more than once across any link, up to the number of routers that share the link. This is due to the forwarding strategy of flooding a packet out all links other than its arriving link, whether or not all the links are part of the shortest-path tree rooted at the sender. This problem is addressed in [6] and also in the following subsection. To distinguish the basic flooding form of reverse path forwarding from later refinements, we refer to it as *reverse path flooding* or *RPF*.

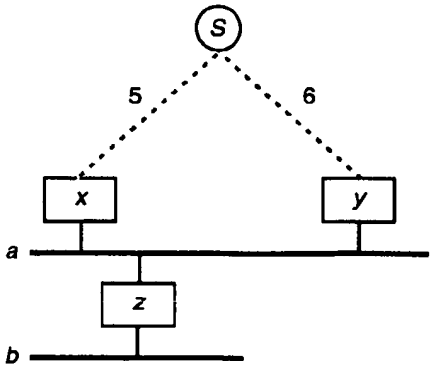


Figure 3: Reverse Path Forwarding Example

5.2 Reverse Path Broadcasting (RPB)

To eliminate the duplicate broadcast packets generated by the RPF algorithm, it is necessary for each router to identify which of its links are "child" links in the shortest reverse-path tree rooted at any given source S. Then, when a broadcast packet originating at S arrives via the shortest path back to S, the router can forward it out only the child links for S.

In [6], Dalal and Metcalfe propose a method for discovering child links which involves each router periodically sending a packet to each of its neighbors, saying, "You are my next hop to these destinations." We propose a different technique for identifying child links which uses only the information contained in the distance-vector routing packets normally exchanged between routers.

The technique involves identifying a single "parent" router for each link, relative to each possible source S. The parent is the one with the minimum distance to S. In case of a tie, the router with the lowest address (arbitrarily) wins. Over each of its links, a particular router learns each neighbor's distance to every S—that is the information conveyed in the periodic routing packets. Therefore, each router can independently decide whether or not it is the parent of a particular link, relative to each S. (This is the same technique as used to select "designated bridges" in Perlman's spanning tree algorithm for LAN bridges [23], except that we build multiple trees, one for each possible source.)

How this works can be seen in the internetwork fragment illustrated in Figure 3. In this example, three routers x, y and z are attached to a LAN a. Router z is also connected to a leaf LAN b. The dashed lines represent the shortest paths from x and from y to a particular source of broadcast packets S, somewhere in the internetwork. The distance from x to S is 5 hops and the distance from y to S is 6 hops. Router z is also 6 hops from S, via x.

To understand the problem being solved, first consider what happens under the basic RPF algorithm. Both x and y receive a broadcast from S over their shortest-path links to S, and both of them forward a copy onto LAN a. Therefore, any hosts attached to a receive duplicate copies of all packets broadcast from S. Router z, however, will forward only one of the copies, the one from x, onto LAN b, because x is z's *next-hop-address* for S.

Now consider how the parent-selection technique solves the problem. All three routers, x, y, and z, periodically send distance-vector routing packets across LAN a, reporting their distance to every destination. From these packets, each of them learns that x has the shortest distance to S. Therefore, only x adopts LAN a as a child link, relative to S; y no longer forwards superfluous broadcasts from S onto LAN a.

If both x and y had a distance of 5 hops to S, the one with the lowest address (say x) would be the parent of LAN a. Note that, in this case, z might choose either x or y as its *next-hop-address* to S. In some implementations of distance-vector routing, z might even alternate between using x and using y to reach S, in order to spread packet traffic over multiple, equally-short paths. However, for the purpose of reverse-path forwarding, every router has to choose a *single* shortest reverse path for each source S. The tie-breaking scheme for parent selection implies that a router with multiple shortest-path routes to S should use the one whose *next-hop-address* is the lowest when deciding whether or not to forward a broadcast from S. Thus, in the example, z forwards broadcasts onto LAN b only if they come from x.

The parent-selection technique for eliminating duplicates requires that one additional field, *children*, be added to each routing table entry. *Children* is a bit-map with one bit for each incident link. The bit for link l in the entry for *destination* is set if l is a child link of this router for broadcasts originating at *destination*.

We call this variant of the algorithm *reverse path broadcasting* or *RPB* because it provides a clean (i.e., no duplicates) broadcast to every link in the internetwork, assuming no transmission errors or topology disruptions.

5.3 Truncated Reverse Path Broadcasting (TRPB)

The RPF and RPB algorithms implement shortest-path broadcasting. They can be used to carry a multicast packet to all links in an internetwork, relying on host address filters to protect the hosts from receiving unwanted multicasts. In a small internetwork with infrequent multicasting, this may be an acceptable approach, just as link-layer bridges that send multicast packets everywhere are acceptable to some. However, as in the

case of large extended LANs, it is desirable in large internetworks to conserve network and router resources by sending multicast packets only where they are wanted. This requires that hosts inform the routers of their group memberships.

To provide shortest-path multicast delivery from source S to members of group G, the shortest-path broadcast tree rooted at S must be pruned back to reach only as far as those links that have members of G. This could be accomplished by requiring members of G to send membership reports back up the broadcast tree towards S, periodically; branches over which no membership reports were received would be deleted from the tree. Unfortunately, this would have to be done separately for every group, over every broadcast tree, resulting in reporting bandwidth and router memory requirements on the order of the total number of groups times the total number of possible sources.

In this subsection, we describe an alternative in which only non-member *leaf* networks are deleted from each broadcast tree. It has modest bandwidth and memory requirements and is suitable for internetworks in which leaf network bandwidth is a critical resource. The next subsection addresses the problem of more radical pruning.

For a router to forgo forwarding a multicast packet over a leaf link that has no group members, the router must be able to (1) identify leaves and (2) detect group membership. Using the algorithm of the previous subsection, a router can identify which of its links are *child* links, relative to a given source S. Leaf links are simply those child links that no other router uses to reach S. (Referring back to Figure 3, LAN b is an example of a leaf link for the broadcast tree rooted at S.) If we have every router periodically send a packet on each of its links, saying, "This link is my next hop to these destinations," then the parent routers of those links can tell whether or not the links are leaves, for each possible destination. In the example, router z would periodically send such a packet on LAN a, saying, "This link is my next hop to S". Hence, router x, the parent of LAN a, would learn that LAN a is not a leaf, relative to S.

Some implementations of distance-vector routing already implicitly convey this next hop information in their normal routing packets, by claiming a distance of *infinity* for all destinations reached over the link carrying the routing packet. This is done as part of a technique known as *split-horizon* which helps to reduce route convergence time when the topology changes [13]. In those cases where the next hop information is not already present, it is necessary only to add one extra bit to each of the (*destination, distance*) pairs in the routing packets. The bits identify which destinations are reached via the link on which the routing packet is being sent.

In the routing tables, another bit-map field, *leaves*, is added to each entry, identifying which of the *children* links are leaf links.

Now that we can identify leaves, it remains for us to detect whether or not members of a given group exist on those leaves. To do this, we have the hosts periodically report their memberships. We can use the membership reporting algorithm described in Section 4, in which each report is locally multicast to the group that is being reported. Other members of the same group on the link overhear the report and suppress their own. Consequently, only one report per group present on the link is issued every reporting interval. There is no need for a very small reporting interval, because it is generally not important to quickly detect when all the members of a group on a link have departed from the group; it just means that packets addressed to that group may be delivered to the link for some time after all the members have left.

The routers then keep a list, for each incident link, of which groups are present on that link. If the lists are stored as hash tables, indexed by group address, the presence or absence of a group may be determined quickly, regardless of the number of groups present. The reverse path forwarding algorithm now becomes: if a multicast packet from S to G arrives from the *next-hop-address* for S, forward a copy out all child links for S except leaf links which have no members of G.

To summarize the costs of this algorithm, which we call *truncated reverse path broadcasting* or *TRPB*:

- It has a storage cost in each router of a few bits added to every routing table entry plus a group list for each of the router's links. The group lists should be sized to accommodate the maximum number of groups expected to be present on a single link (although temporary overflows of a group list may safely be handled by temporarily treating the corresponding link as a non-leaf, forwarding *all* multicast packets onto the link).
- It has a bandwidth cost on each link of one membership report per group present per reporting interval. The membership reports are very small, fixed-length packets, and the reporting interval may reasonably be on the order of minutes.
- The bandwidth cost of conveying next hop information in the routing packets is typically zero, either because the split horizon technique is used, or because an unused bit can be stolen from the existing (*destination, distance*) pairs to carry that information.

5.4 Reverse Path Multicasting (RPM)

As mentioned in the previous subsection, pruning the shortest-path broadcast trees by sending membership reports towards each multicast source results in an explosion of reporting traffic and router memory requirements. In a large internetwork, we would not expect every possible source to send multicast packets to every existing group, so the great expense of pruning every possible multicast tree would be wasted. We would prefer, then, to prune only those multicast trees that are actually in use.

Our final variation on the reverse path forwarding strategy provides *on-demand pruning* of shortest-path multicast trees, as follows. When a source *first* sends a multicast packet to a group, it is delivered along the shortest-path broadcast tree to all links except non-member leaves, according to the TRPB algorithm. When the packet reaches a router for whom all of the child links are leaves and none of them have members of the destination group, a *non*-membership report (NMR) for that (*source, group*) pair is generated and sent back to the router that is one hop towards the source. If the one-hop-back router receives NMRs from all of its child routers (that is, all routers on its child links that use those links to reach the source of the multicast), and if its child links also have no members, it in turn sends an NMR back to its predecessor. In this way, information about the absence of members propagates back up the tree along all branches that do not lead to members. Subsequent multicast packets from the same source to the same group are blocked from travelling down the unnecessary branches by the NMRs sitting in intermediate routers.

A non-membership report includes an *age* field, initialized by the router that generates the report, and counted up by the router that receives the report. When the age of an NMR reaches a threshold, T_{maxage}, it is discarded. The NMRs generated at the leaves start with age zero; NMRs generated by intermediate routers, as a consequence of receiving NMRs from routers nearer the leaves, start with the maximum age of all of the subordinate NMRs. Thus, any path that is pruned by an NMR will rejoin the multicast tree after a period of T_{maxage}. If, at that time, there is still traffic from the same source to the same group, the next multicast packet will trigger the generation of a new NMR, assuming there is still no member on that path.

When a member of a new group on a particular link appears, it is desirable that that link immediately be included in the trees of any sources that are actively sending to that group. This is done by having routers remember which NMRs they have sent and, if necessary, send out cancellation messages to undo the effect of the NMRs.

If an NMR is lost in transit, a subtree will remain in the multicast tree unnecessarily, but that will last only until the next multicast packet stimulates generation of another NMR. Loss of a cancellation message is more serious, because a new path will fail to join the tree when it should, and group members on that path will fail to receive multicast packets from that tree for a period of up to up to T_{maxage}. If we require that cancellation messages be positively acknowledged by their receivers, we can afford to have a very long T_{maxage}, which reduces the amount of multicast traffic down unnecessary branches.

This algorithm, which we call *reverse path multicasting* or *RPM*, has the same costs as the TRPB algorithm, plus the costs of transmitting, storing, and processing NMRs and cancellation messages. Those extra costs depend greatly on such factors as the number and locations of multicast sources and of group members, the multicast traffic distributions, the frequency of membership changes, and the internetwork topology. In the worst case, the number of NMRs that a router must store is on the order of the number of multicast sources active within a T_{maxage} period, times the average number of groups they each send to in that period, times the number of adjacent routers. There are a couple of factors that can alleviate these storage requirements:

- All hosts attached to the same link may be treated as a single source of multicasts, as long as a router is able to identify the source link from the source addresses of datagrams, as is the case, for example, with DoD IP addresses [24].
- Multicast datagrams sent with a small time-to-live may expire before reaching many routers, thus avoiding the generation of NMRs in those routers.

We believe that many applications of internetwork multicasting will be able to use TTL scope control effectively, either because they require communication with only a nearby subset of a large group (e.g., when looking for a nearby name server), or because all group members are known to be close to the senders (e.g., when a parallel computation is distributed across computers at a single site). If that is so, and the cost of memory keeps falling, storage space for NMRs should not be a limiting factor in typical distance-vector routing environments (fewer than a hundred links). Bandwidth can also be expended to recover memory, by reducing T_{maxage}. However, experience with real multicast traffic in real internetworks will be needed before recommendations can be made as to router memory sizes, timeout values, or even whether the greater "precision" of the RPM algorithm is worth the extra complexity and overhead, as compared to the simpler TRPB algorithm.

One issue that has not yet been mentioned in this discussion of reverse path forwarding schemes is the effect of topology changes. As explained in [6], reverse path forwarding can cause packets to be duplicated or lost if routing tables change while the packets are in transit. Since we require only datagram reliability, occasional packet loss or duplication is acceptable; hosts are assumed to provide their own end-to-end recovery mechanisms to the degree they require them. Implementations of the RPM algorithm, however, must be careful to take into account any topology changes that might modify the pruned multicast trees. For example, when a router gains a new child link or a new child router, relative to a given multicast source, it must send out cancellation messages for any outstanding NMRs it has for that source, to ensure that the new link or router is included in future multicast transmissions from that source.

6 Link-State Multicast Routing

The third major routing style to be considered is that of link-state routing, also known as "New Arpanet" or "Shortest-Path-First" routing [21]. As well as being used in the Arpanet, the link-state algorithm has been proposed by ANSI as an ISO standard for intra-domain routing [18].

Under the link-state routing algorithm, every router monitors the state of each of its incident links (e.g., up/down status, possibly traffic load). Whenever the state of a link changes, the routers attached to that link broadcast the new state to every other router in the internetwork. The broadcast is accomplished by a special-purpose, high-priority flooding protocol that ensures that every router quickly learns of the new state. Consequently, every router receives information about all links and all routers, from which they can each determine the complete topology of the internetwork. Given the complete topology, each router independently computes the shortest-path spanning tree rooted at itself, using Dijkstra's algorithm [1]. From this tree, it determines the shortest path from itself to any destination, to be used when forwarding packets.

It is straightforward to extend the link-state routing algorithm to support shortest-path multicast routing. Simply have routers include as part of the "state" of a link, a list of groups that have members on that link. Whenever a new group appears, or an old group disappears, from a link, the routers attached to that link flood the new state to all other routers. Given full knowledge of which groups have members on which links, any router can compute the shortest-path multicast tree from any source to any group, using Dijkstra's algorithm. If the router doing the computation falls within the computed tree, it can determine which links it must use to forward copies of multicast packets from the given source to the given group.

To enable routers to monitor group membership on a link, we again use the technique, introduced in Section 4, of having hosts periodically issue membership reports. Each membership report is transmitted as a local multicast to the group being reported, so that any other members of the same group on the same link can overhear the report and suppress their own. Routers monitoring a link detect the departure of a group by noting when the membership reports for that group stop arriving. This technique generates, on each link, one packet per group present per reporting interval.

It is preferable for only one of the routers attached to a link to monitor the membership of that link, thereby reducing the number of routers that can flood membership information about the link. In the link-state routing architecture proposed in [18], this job would fall to the "LAN Designated Router", which already performs the task of monitoring the presence of individual hosts.

As pointed out in Section 5, there is potentially a separate shortest-path multicast tree from every sender to every group, so it would be very expensive in space and processing time for every router to compute and store all possible multicast trees. Instead, we borrow from Section 5.4 the idea of only building trees on demand. Each router keeps a cache of multicast routing records of the form:

(*source, subtree,* (*group, link-ttls*),
(*group, link-ttls*), ...)

Source is the address of a multicast source. *Subtree* is a list of all descendent links of this router, in the shortest-path spanning tree rooted at *source*. *Group* is a multicast group address. *Link-ttls* is a vector of time-to-live values, one for each incident link, specifying the minimum TTL required to reach the nearest descendent member of the group via that link; a special TTL value for *infinity* identifies links that do not lead to any descendent members.

When a router receives a multicast packet, it looks up the source of the packet in its multicast routing cache. If it finds a record, it looks for the destination group in the (*group, link-ttls*) fields. If the group is found, the router forwards the packet out all links for which the minimum required TTL in *link-ttls* is less than or equal to the TTL in the packet header.

If the source record is found, but the destination group is not in the record, the router must compute the outgoing links and corresponding TTLs. To do this, it scans through the links in *subtree*, looking for links that have members of the destination group, and computing

the minimum TTLs required to reach any member links found. The new *group* and *link-ttls* are added to the record and used in the forwarding decision.

Finally, if a record is not found for the source of an incoming multicast packet, the complete shortest-path spanning tree for that source must be computed. From the tree, the subtree of descendents of the router can be identified. The *source* and *subtree* are then installed as a new record in the multicast routing cache. The *link-ttls* for the destination group are also computed as part of computing the full tree, added to the record, and used in the forwarding decision. (A router for whom memory is scarcer than processing power might choose not to store the *subtrees* in the multicast routing cache, and simply recompute the full tree whenever a new group for a particular source is encountered.)

Cache records need not be timed out. When the cache is full, old records may be discarded on a least-recently-used basis. Whenever the topology changes, all cache records are discarded. Whenever a new group appears, or an old group disappears, on a link, all (*group, link-ttls*) fields identifying that group are removed from the cache.

Like the RPM algorithm described in the previous section, the costs of this algorithm are very dependent on the internetwork multicast traffic patterns. Assuming that there are generally fewer groups present on a single LAN than there are individual hosts, the bandwidth required for group link state packets should be no more than that required for "End System" link state packets, in the proposed ANSI routing scheme [18]. The same is true of the memory needed in the routers to hold the link membership information. The major costs of the algorithm are in the memory required to store the multicast routing cache records and the processing requirements of computing the multicast trees. Assuming that most multicast packets are required to traverse a small percentage of the routers in the internetwork, this algorithm requires less storage space than the RPM algorithm, because storage is consumed only in those routers that must be traversed, rather than in those that must *not* be traversed.

One possible drawback of this algorithm is the additional delay that may be imposed on the first multicast packet transmitted from a given source—at each hop, the routers must compute the full tree for that source before they can forward the packet. The complexity of the tree computation is of the order of the number of the links in the internetwork (for sparsely-connected interworks); decomposing a large internetwork into routing subdomains, as proposed in the ANSI scheme, is an effective way of controlling the number of links within any domain.

7 Hierarchical Multicast Routing

All of the algorithms discussed so far are appropriate for a single routing domain, in which all routers are running the same algorithm. Large internetworks often span *multiple* routing domains. For example, a LAN that is part of a distance-vector routing environment may actually be an extended LAN containing spanning-tree bridges, or one "link" in a link-state routing environment may actually be an entire internetwork using distance-vector routing. Such hierarchical composition—treating one routing domain as a single link in a higher-level routing domain—has many advantages. It reduces the amount of topology information any one router has to maintain, thereby improving scaleability [19]; it accommodates different technologies for which different routing strategies are appropriate; and it allows different organizations to choose the routing style that best fits their needs, while still interoperating with other organizations.

All of the multicast routing algorithms we have proposed may be used to route multicast packets between "links" that happen to be entire routing subdomains, provided that those subdomains meet our requirements for links. Section 3 identifies the two generic types of links assumed by the multicast algorithms: point-to-point links and multi-access links. A subdomain may be treated as a point-to-point link if it used only for pairwise communication between two routers or between a router and a single host. Alternatively, a subdomain may be treated as a multi-access link if it satisfies the following property:

- If any host or superdomain router attached to the subdomain sends a multicast packet addressed to group G into the subdomain, it is delivered (with high probability) to all hosts that are members of G *plus* all superdomain routers attached to the subdomain, subject to the packet's time-to-live (TTL).

In addition, if the superdomain multicast routing protocol does *not* use the approach of delivering every multicast packet to every link, it must be possible for the superdomain routers to monitor the group membership of hosts attached to the subdomain. This may be done using the membership reporting protocol described in the previous sections, or via some other, subdomain-specific, method.

The above property is required of a subdomain when using our algorithms as superdomain multicast routing protocols. Looking at it from the other side, when using our algorithms as *subdomain* multicast routing protocols beneath an arbitrary superdomain protocol, we find that we do not quite satisfy the above property for subdomains. We must extend our algorithms to include all superdomain routers as members of every group, so that they may receive all multicast packets sent within the subdomain. This is accomplished simply by defining

within the subdomain a special "wild-card" group that all superdomain routers may join; the changes to each algorithm to support wild-card groups are straightforward.

8 Related Work

A variety of algorithms for multicast routing in store-and-forward networks are described by Wall [26], with emphasis on algorithms for constructing a single spanning tree that provides low average delay, thereby striking a balance between opposing goals of low delay and low network cost.

Frank, Wittie and Bernstein [10] provide a good survey of multicast routing techniques that can be used in internetworks, rating each according to such factors as delay, bandwidth, and scaleability.

Sincoskie and Cotton [25] propose a multicast routing algorithm for link-layer bridges which supports a type of group in which all senders must also be members of the group. Such groups are acceptable for some applications, such as computer conferencing, but are not well suited to the common client/server type of communication where the (client) senders are generally not members of the (server) group and should not receive packets sent to the group.

9 Conclusions

We have proposed a number of algorithms for routing multicast datagrams in internetworks and extended LANs. The goal of each algorithm is to provide a multicast service that is as similar as possible to LAN multicasting, so that applications that currently benefit from LAN multicasting may be moved to a multiple-network environment with little or no change. In particular, we have concentrated on *low delay* multicasting, in order to minimize the effect of going from the LAN environment to a store-and-forward environment.

Different multicast routing algorithms were developed as extensions to three different styles of *unicast* routing: the single-spanning-tree routing of extended LAN bridges, and the distance-vector and link-state routing commonly used in internetworks. These different routing styles lead to significantly different multicast routing strategies, each exploiting the particular protocols and data structures already present.

For most of the algorithms, the additional bandwidth, memory and processing requirements are not much greater than those of the underlying unicast routing algorithm. In the case of distance-vector routing, we presented a range of multicast routing algorithms based on Dalal and Metcalfe's reverse path forwarding scheme, providing increasing "precision" of delivery (flooding, broadcasting, truncated broadcasting and multicasting) at a cost of increasing amounts of routing overhead.

In spite of the wide difference in multicast routing strategies, all except the flooding and broadcasting variants impose the same requirement on hosts: a simple membership reporting protocol which takes good advantage of multicasting to eliminate redundant reports. Thus, the same host protocol implementation may be used without change in a variety of different multicast routing environments.

Finally, we have shown how different routing domains using these or other multicast routing protocols may be combined to extend multicasting across a large, hierarchical internetwork.

We have implemented the host membership reporting protocol in the 4.3BSD UNIX kernel as the first step in an experiment with internetwork multicasting of DoD IP datagrams [7], and implementations of both the reverse path multicast (RPM) and the link-state multicast routing algorithms are under way. From these implementations, we plan to derive detailed specifications for each of the multicast routing algorithms, and to start gathering measurements of multicast traffic patterns and their effect on routing overhead, for a variety of distributed multicast applications, such as computer conferencing, name binding, and network management. Once we get a better idea of multicast "workloads", we hope to provide stronger criteria for choosing among the various multicast routing algorithms.

Acknowledgements

The idea of applying the membership reporting strategy to the extended LAN environment was suggested by David Cheriton. David Waitzman pointed out how the link-state multicasting algorithm could take into account TTL scope control. They, as well as Bruce Hitson, Cary Gray, and an anonymous reviewer, provided many other helpful comments on an earlier draft of this paper. The DARPA Internet task force on end-to-end protocols, chaired by Bob Braden, has encouraged and contributed to the development of these ideas.

This work was sponsored in part by the Defense Advanced Research Projects Agency under contract N00039-84-C-0211, and by Digital Equipment Corporation.

References

[1] A. V. Aho, J. E. Hopcroft, and J. D. Ullman. *Data Structures and Algorithms*. Addison-Wesley, Reading, Mass., 1983. Dijkstra's algorithm.

[2] R. E. Bellman. *Dynamic Programming*. Princeton University Press, Princeton, N.J., 1957.

[3] D. R. Boggs. *Internet Broadcasting*. PhD thesis, Electrical Engineering Dept., Stanford University, January 1982. Also Tech. Rep. CSL-83-3, Xerox PARC, Palo Alto, Calif.

[4] D. R. Boggs, J. F. Shoch, E. A. Taft, and R. M. Metcalfe. PUP: an internetwork architecture. *IEEE Transactions on Communications*, COM-28(4):612–624, April 1980.

[5] D. R. Cheriton and W. Zwaenepoel. Distributed process groups in the V kernel. *ACM Transactions on Computer Systems*, 3(2):77–107, May 1985.

[6] Y. K. Dalal and R. M. Metcalfe. Reverse path forwarding of broadcast packets. *Communications of the ACM*, 21(12):1040–1048, December 1978.

[7] S. E. Deering. *Host Extensions for IP Multicasting*. RFC 1054, SRI Network Information Center, May 1988.

[8] Digital Equipment Corporation, Intel Corporation, and Xerox Corporation. The Ethernet: a local area network; data link layer and physical layer specifications, version 1.0. *Computer Communications Review*, 11(3):20–66, September 1980.

[9] L. R. Ford Jr. and D. R. Fulkerson. *Flows in Networks*. Princeton University Press, Princeton, N.J., 1962.

[10] A. J. Frank, L. D. Wittie, and A. J. Bernstein. Multicast communication on network computers. *IEEE Software*, 2(3):49–61, May 1985.

[11] J. Hart. Extending the IEEE 802.1 MAC bridge standard to remote bridges. *IEEE Network*, 2(1):10–25, January 1988.

[12] W. R. Hawe, M. F. Kempf, and A. J. Kirby. The extended local area network architecture and LANBridge 100. *Digital Technical Journal*, (3):54–72, September 1986.

[13] C. Hedrick. *Routing Information Protocol*. RFC (in preparation), SRI Network Information Center, November 1987.

[14] R. Hinden and A. Sheltzer. *The DARPA Internet Gateway*. RFC 823, SRI Network Information Center, September 1982.

[15] IEEE Computer Society. Standards for local area networks: logical link control. ANSI/IEEE Standard 802.2-1985 (ISO/DIS 8802/2), 1985.

[16] International Business Machines Corp. *Technical Reference PC Network*. document 6322916.

[17] International Organization for Standardization (ISO). *Draft International Standard 8473, Protocol for Providing the Connectionless-Mode Network Service*. March 1986.

[18] Secretariat USA (ANSI) ISO TC97 SC6. *Intermediate System to Intermediate System Intra-Domain Routing Exchange Protocol*. November 1987.

[19] L. Kleinrock and F. Kamoun. Hierarchical routing for large networks; performance evaluation and optimization. *Computer Networks*, 1:155–174, 1977.

[20] S. J. Leffler, R. S. Fabry, W. N. Joy, P. Lapsley, S. Miller, and C. Torek. An advanced 4.3BSD interprocess communication tutorial. In *Unix Programmer Supplementary Documents, Part 2*, University of California, Berkeley, Ca., April 1986.

[21] J. M. McQuillan, I. Richer, and E. C. Rosen. The new routing algorithm for the ARPANET. *IEEE Transactions on Communications*, COM-28(5):711–719, May 1980.

[22] J. M. McQuillan and D. C. Walden. The ARPANET design decisions. *Computer Networks*, 1, August 1977.

[23] R. Perlman. An algorithm for distributed computation of a spanning tree in an extended LAN. In *Proc. 9th Data Communications Symposium*, pages 44–53, ACM/IEEE, September 1985.

[24] J. Postel. *Internet Protocol*. RFC 791, SRI Network Information Center, September 1981.

[25] W. D. Sincoskie and C. J. Cotton. Extended bridge algorithms for large networks. *IEEE Network*, 2(1):16–24, January 1988.

[26] D. W. Wall. *Mechanisms for Broadcast and Selective Broadcast*. PhD thesis, Electrical Engineering Dept., Stanford University, June 1980. Also Tech. Rep. 190, Computer Systems Lab., Stanford.

[27] Xerox Corporation. *Internet Transport Protocols*. XSIS 028112, Xerox, Stamford, Connecticut, December 1981.

MBONE, the Multicast BackbONE

Mike Macedonia and Don Brutzman

Naval Postgraduate School

Introduction.

The joy of science is in the discovery. It was a year ago when we heard that the JASON Project, an underwater exploration educational program supported by Woods Hole Oceanographic Institution, was showing live video over the Internet from an underwater robot in waters off Baja, Mexico. Our group here at the Naval Postgraduate School (NPS) furiously tried to figure out how to receive that video signal. We worked for several days to gather the right equipment, contact the appropriate network managers, and get hardware permissions from local bureaucrats, only to find that an antenna uplink on the JASON support ship had flooded a few hours before we became operational. Despite this disappointment we were not discouraged because we had discovered how to use the Internet's most unique network, MBONE.

MBONE stands for Multicast Backbone, a virtual network that has been in existence for about three years. The network originated from an effort to multicast audio and video from the Internet Engineering Task Force (IETF) meetings. MBONE today is used by several hundred researchers for developing protocols and applications for group communication. Multicast is used because it provides one-to-many and many-to-many network delivery services for applications such as videoconferencing and audio that need to communicate with several other hosts simultaneously.

Multicast networks.

Multicasting has existed for several years on local area networks such at Ethernet and FDDI. However, with Internet Protocol (IP) multicast addressing at the network layer the service group communication can be established across the Internet. IP multicast addressing is an Internet standard developed by Steve Deering (Request For Comment RFC-1112) and is supported by a number of workstation vendors, including Sun and Silicon Graphics Inc. Categorized officially as an IP Class D address, an IP multicast address is mapped to the underlying hardware multicast services of a local area network.

The reason that MBONE is a virtual network is that it shares the same physical media as the Internet, though it must use a parallel system of routers that can support multicast (e.g. dedicated workstations running with modified kernels and multiple interfaces) augmented with "tunnels". Tunneling is a scheme to forward multicast packets among the islands of MBONE subnets through Internet IP routers which typically do not support IP multicast. This is done by encapsulating the multicast packets inside regular IP packets.

Bandwidth.

The key to understanding the constraints of MBONE is thinking about bandwidth. The reason why a multicast stream is bandwidth-efficient is that one packet can touch all workstations on a network. Thus a 125 Kbps video stream (1 frame/second) uses the same bandwidth whether it is received by one

workstation or twenty. That is good. However that same multicast packet is prevented from crossing network boundaries such as routers or bridges. The reasons for this restriction are religious and obvious: if a multicast stream which can touch every workstation could jump from local network to local network, then the entire Internet would quickly become saturated by such streams. That is very bad! Thus the MBONE scheme encapsulates multicast streams into unicast packets which can be passed as regular Internet protocol packets along a virtual network of dedicated multicast routers (mrouters) until they reach the various destination local area networks. The use of dedicated mrouters segregates MBONE packet delivery, protecting standard network communications such as mail and telnet from MBONE experiments and failures. Once properly established, an mrouter needs little or no attention. Given this robust distribution scheme, responsible daily use of the MBONE network consists only of making sure you don't overload your local or regional bandwidth capacity.

Networking details.

When a host on an MBONE-equipped subnet establishes or joins a group it announces that event via the Internet Group Management Protocol (IGMP). The multicast router on the subnet forwards it the other routers in the network. MBONE sessions use a tool developed by Van Jacobson of Lawrence Berkeley Laboratories called sd (session directory) to display the announcements by multicast groups. sd is also used for launching multicast applications and for automatically selecting an unused address for any new groups.

Groups are disestablished when everyone leaves, freeingup the IP multicast address for reuse. The routers occasionally poll hosts on the subnets to determine if any are still group members. If there is no reply by a host, the router stops advertising that hosts group membership to the other multicast routers.

The routing protocols for MBONE are still immature andtheir ongoing design is a central part of this network experiment. Most MBONE routers employ the Distance Vector Multicast Routing Protocol (DVMRP) which is commonly considered inadequate for rapidly changing network topologies because routing information propagates too slowly. A multicast extension to the Open Shortest Path (MOSPF) link-state protocol has been proposed by John Moy of Proteon Inc. that addresses this problem. However, with both protocols each router must compute a source tree for each participant in a multicast group. MBONE is currently small enough that this restriction is not a problem. However, for a large network with constantly changing group memberships such routing techniques are expected to be computationally inefficient.

Topology and Scheduling.

The MBONE topology and the scheduling of multicast sessions must be actively managed by the MBONE community to minimize congestion. Approximately 400 sites worldwide are currently MBONE members. MBONE protocol developers are currently experimenting with automatically pruning and grafting subtrees, but for the most part uses truncated broadcasts to the leaf routers. The truncation is based on the setting for the time-to-live (ttl) field in a packet which is decremented each time the packet passes though a router. A ttl value of 16 would limit multicast to a campus, as opposed to a value of 100 which might send it to every subnet on the entire MBONE (about thirteen countries).

These issues can have a major impact on network performance. For example, a default video stream consumes about 128 Kbps (kilobits per second) of bandwidth, which is almost 10 percent of a T1 line (a common site-to-site link on the Internet). Several simultaneous high-bandwidth sessions might easily saturate network links and routers. This problem is compounded by the fact that general purpose

workstation routers typically used by the MBONE are normally not as fast or as robust as the dedicated hardware routers used in most of the Internet.

Protocols.

The magic of MBONE is that teleconferencing can be done in the hostile world of the Internet where variable packet delivery delays and limited bandwidth play havoc with applications that require some real-time guarantees. It is worth noting that only a few years ago puting audio and video across the Internet was considered impossible. Development of effective multicast protocols disproved that widespread opinion. In this respect MBONE is like the proverbial talking dog: it's not so much what the dog has to say, it's more that the dog can talk at all that is amazing.

In addition to the multicast protocols, MBONE applications are using the Real Time Protocol (RTP) on top of User Datagram Protocol (UDP) and IP. RTP is being developed by the Audio-Video Transport Working Group within the IETF. RTP provides timing and sequencing services; permitting the application to adapt and smooth out network-induced latencies and errors. The end result is that even with a time-critical application like an audio tool, participants normally perceive conversations as if they are in real-time, even though there is actually a small buffering delay to synchronize and sequence the arriving voice packets. Protocol development continues. Although operation is usually robust, many aspects of MBONE are still considered experimental.

Data Compression.

Another aspect of this research is the need to compress a variety of media and to provide privacy through encryption. Several techniques to reduce bandwidth include Joint Photographic Experts Group (JPEG) compression and the ISO standard H.261 for video. Visually this translates to velocity compression: rapidly changing screen blocks are updated much more frequently than slowly changing blocks. Encodings for audio include Pulse Coded Modulation (PCM) and GSM. Ouside of the concerns for real-time delivery, audio is a difficult media for the MBONE and teleconferencing in general because of the need to balance signal levels for all the parties who may have different audio processing hardware (e.g. microphones and amplifiers). Audio also generates lots of relatively small packets, which are the bane of network routers.

Application Tools.

Besides basic networking technology, MBONE researchers are developing new applications that typify many of the goals associated with an "information superhighway." Video, audio, and a shared drawing whiteboard are the principal applications, provided by software packages called nv (net video), vat (visual audio tool) and wb (whiteboard). The principal authors of these tools are Ron Frederick of Xerox Palo Alto Research Center (PARC) for nv and Van Jacobson of Lawrence Berkeley Laboratory for vat and wb. Each of these programs is available in compilable or executable form without charge from various anonymous ftp sites on the Internet. Working versions are available for Sn, SGI and VMS architectures with ports in progress for HP-UX and Macintosh. No DOS, OS-2 or Windows versions are currently available although ported tools can be found for 386 boxes running the (free) 386bsd Unix. Pointers to all public application tools are included in the FAQ.

Additional tools are also available or under development. Winston Dang of the University of Hawaii has created imm (Image Multicaster Client), a low-bandwidth image server. It is typically used to provide live

images of planet Earth from geostationary satellites at half-hour intervals in either visible or infrared (IR) spectra. Article author Mike Macedonia has ported the IEEE Distributed Interactive Simulation (DIS) protocol to enable live interaction between virtual worlds as an MBONE communications tool. Other researchers are experimenting with using graphics workstation windows as image drivers. Future network news distributions may be multicast to reduce overall network loading and speed news delivery.

Events.

Many of the most exciting events on the Internet appear first on MBONE. Perhaps the most popular is NASA Select, the NASA in-house cable channel broadcast during space shuttle missions. Be warned that this might be a work stopper! It is hard to describe the excitement of seeing one astronaut hold another astronaut by the boots to repair a satellite, all live from 150 miles above the earth. Conferences on supercomputing, Internet Engineering Task Force, scientific visualization and many other topics have appeared, often accompanied by directions on how to download PostScript copies of presented papers and slides from anonymous ftp sites. Radio Free VAT is a community radio station whose DJ's sign up for air time via an automated server (vat-radio-request@elxr.jpl.nasa.gov). Xerox PARC occasionally broadcasts lectures by distinguished speakers. Internet Talk Radio (Carl Malamud, info@radio.com) has presented talks by Larry King and "Geek of the Week." Remote learning has brought expertise over long distances and multiplied training benefits. Default MBONE audio and video channels are available for new users, experimentation and advice from more experienced users.

Groupwork on groupware.

The MBONE community is active and open. Work on tools, protocols, standards, applications and events is very much a cooperative and international effort. Feedback and suggestions are often relayed to the entire MBONE mailing list (as an example, this article was proofed by that group). Cooperation is essential due to the limited bandwidth of many networks, in particular transoceanic links. So far no hierarchical scheme has been necessary for resolving potentially contentious issues such as topology changes or event scheduling. Distributed problem solving and decision making has worked on a human level just as successfully as on the network protocol level. Hopefully this decentralized approach will continue to be successful even in the face of rapid addition of new users.

Cost of admission.

The first thing you need to get on MBONE is the willingness to study and learn how to use these new and fast-moving tools. The second thing you need is bandwidth. Here at NPS we run MBONE tools on workstations connected via Ethernet (10 Mbps). Off-campus links are via T1 lines (1.5 Mbps). We have found that bandwidth capacities lower than T1 will result in network crashes and thus appear unsuitable for MBONE.

Given adequate network bandwidth, you now need a designated MBONE network administrator. It typically takes one to three weeks for a network-knowledgeable person working part-time to establish MBONE at a new site. Setup is not for the faint of heart, but all of the tools are documented and help is available from the MBONE list. Read the Frequently Asked Questions (FAQ) a few times and ensure that software tools and multicast- compatible kernels are available for your target workstations. Subscribe to the mail list in advance so that you will be able to ask questions and receive answers. Figure 1 shows the various worldwide MBONE list subscription request addresses. After subscribing, read the FAQ again.

To receive multicast packets on your local area network, you will need to configure an mrouter which strips off packet encapsulation. This can be a dedicated router. A more popular approach is to take an old slow cast-off workstation and equip it with two Ethernet cards. Two network cards are needed, one to receive the upstream tunnel, and the other to distribute downstream multicast packets. Obtain and load the application software tools. You are now ready to put multicast on your local area network.

Note that these tools can also work in isolation between workstations on a single local area network without any mrouter. We recommend that you test the application tools locally in advance (before going through the mrouter effort) to see if they are compatible and match your expectations.

Once you are connected, pass along any lessons learned to the tool authors or the MBONE list. Later show your overall network site administrator something spectacular on MBONE (such as a live spacewalk) and then make sure that your site is programming funds to increase your network bandwidth. Demands on network bandwidth are significant and getting more critical. You might consider Tengdin's First (and Only) Law of Telecommunications: "The jump from zero to whatever baud rate is the most important jump you can make. After that everyone always wnts to go straight to the speed of light."

Caveats.

Some problems still exist and a lot of work is still in progress. The audio interface takes coaching and practice. Leaving your microphone on by mistake may override everyone else since only one person can talk at a time. You will need a video capture board in your workstation to transmit video, but no special hardware isneeded to receive video. One frame per second video seems pretty slow (standard video is 30 frames per second), but in practice it is surprisingly effective when combined with phone-quality voice. One user blasting a high-bandwidth video signal (greater than 125 Kbps) can cause severe and widespread network problems. Controls on access to tools are rudimentary and security is minimal; for example, a local user might figure out how to listen through your workstation mike (unless you unplug it). Audio broadcast preparations are often just as involved as video broadcast preparations. Network monitoring tools are not yet convenient to use. Internet bandwidth is still inadequate for MBONE in many countries. On one occasion a local topology change at our school caused a feedback loop that overrode the NASA Select audio track. Although plenty of people were willing to point out the symptoms of our error (!) it was not possible for the rest of the network to cut off the offending workstation cleanly. More situations will undoubtedly occur as the MBONE developers and users learn more and continue to improve the tools. Expect to spend some time if you want to be an MBONE user.

The future.

It is not every day that someone says to you "Here is a multimedia television station that you can use to broadcast from your desktop to the world." These are powerful concepts and powerful tools that tremendously extend our ability to communicate and collaborate. These tools are already changing the way people work and interact on the net. See you later!

Further reading

1. Casner, Steve. "Frequently Asked Questions (FAQ) on the Multicast Backbone," 6 May 1993, available via anonymous ftp from venera.isi.edu:/mbone/faq.txt

2. Casner, Steve and Schulzrinne, Henning. "RTP: A Transport Protocol for Real-Time Applications,"

IETF Draft, 20 October 1993.

3 Casner, Stephen and Deering, Stephen. "First IETF Internt Audioast," ACM SIGCOMM Computer Communication Review, San Diego California, July 1992, pp. 92-97.

4. Comer, Douglas E. Internetworking with TCP/IP, volume I, Prentice-Hall, New Jersey, 1991.

5. Deering, Stephen. "Host Extensions for IP Multicastig", RFC 1112, August 1989.

6. Deering, Stephen. "MBONE-The Multicast Backbone," CERFnet Seminar, 3 March 1993.

7. Moy, John. "Multicast Extensions to OSPF," IETF Draft, July 1993.

8. Perlman, Radia. Interconnections: Bridges and Routers, Addison-Wesley, New York, 1993, p. 258.

9. Curtis, Pavel, mbone map, available via anonymous ftp from
parcftp.xerox.com:pub/net-research/mbone-map-big.ps

The final article will be available electronically as taurus.cs.nps.navy.mil:pub/mbmg/mbone.hottopic.ps

Major Michael R. Macedonia USA is a Ph.D. student. Lieutenant Commander Donald P. Brutzman USN is an instructor and Ph.D. candidate. Both can be reached at Computer Science Department, Naval Postgraduate School, Monterey California USA 93943-5000; e-mail addresses are macedoni@cs.nps.navy.mil and brutzman@cs.nps.navy.mil.

This page includes all Drafts and RFCs which are related to IP-Multicast. The page is updated (Last Update: 13.02.2003)on a regular basis. Please contact me if you encounter any problems. Drafts and RFCs which do not belong to one of the main Multicast related Working groups are listed at the end with a detailed description. Currently this list goes back until July 2002 (please have in mind, that drafts expire after six month). Old version of drafts are kept on this list.

BGPM, IDMR, MAGMA, MALLOC, MBONED, MOSPF, MSDP, MSEC, PIM, RMT, SSM

other related IETF drafts

BGMP - Border Gateway Multicast Protocol (BGMP@IETF)

- Drafts:
  - Border Gateway Multicast Protocol (BGMP): Protocol Specification (draft-ietf-bgmp-spec-03.txt)
- RFCs:
  - <none>

Back

IDMR - Inter Domain Multicast Routing WG (IDMR@IETF)

- Drafts:
  - Distance Vector Multicast Routing Protocol (draft-ietf-idmr-dvmrp-v3-10.txt)
  - Multicast Router Discovery
    - draft-ietf-idmr-igmp-mrdisc-09.txt
    - draft-ietf-idmr-igmp-mrdisc-09.txt
  - Distance Vector Multicast Routing Protocol Applicability Statement (draft-ietf-idmr-dvmrp-v3-as-00.txt)
- RFCs:
  - RFC 1949: Scalable Multicast Key Distribution
  - RFC 2117: Protocol Independent Multicast-Sparse Mode (PIM-SM): Protocol Specification
  - RFC 2201: Core Based Trees (CBT) Multicast Routing Architecture
  - RFC 2189: Core Based Trees (CBT version 2) Multicast Routing: Protocol Specification
  - RFC 2236: Internet Group Management Protocol, Version 2
  - RFC 2362: Protocol Independent Multicast-Sparse Mode (PIM-SM): Protocol Specification
  - RFC 2715: Interoperability Rules for Multicast Routing Protocols
  - RFC 2932: IPv4 Multicast Routing MIB
  - RFC 2933: Internet Group Management Protocol MIB
  - RFC 2934: Protocol Independent Multicast MIB for IPv4
  - RFC 3376: Internet Group Management Protocol, Version 3

Back

MAGMA - Multicast & Anycast Group Membership (MAGMA@IETF)

- Drafts:
  - Using IGMPv3 and MLDv2 For Source-Specific Multicast (draft-holbrook-idmr-igmpv3-ssm-03.txt)
  - IGMP/MLD-based Multicast Forwarding ('IGMP/MLD Proxying') (draft-ietf-magma-igmp-proxy-01.txt)
  - Multicast Source Notification of Interest Protocol (MSNIP) (draft-ietf-magma-msnip-01.txt)
  - Multicast Router Discovery SSM Range Option (draft-ietf-magma-mrdssm-01.txt)
  - Socket Interface Extensions for Multicast Source Filters (draft-ietf-magma-msf-api-03.txt)
  - Multicast Listener Discovery Version 2 (MLDv2) for IPv6 (draft-vida-mld-v2-05.txt)
  - IGMPv3/MLDv2 and Multicast Routing Protocol Interaction
    - draft-ietf-magma-igmpv3-and-routing-04.txt
    - draft-ietf-magma-igmpv3-and-routing-03.txt
  - Considerations for IGMP and MLD snooping switches
    - draft-ietf-magma-snoop-04.txt
    - draft-ietf-magma-snoop-03.txt
    - draft-ietf-magma-snoop-02.txt
  - Internet Group Management Protocol MIB (draft-ietf-magma-rfc2933-update-00.txt)
  - Source Address Selection for Multicast Listener Discovery Protocol (RFC2710)
    - draft-ietf-magma-mld-source-04.txt
    - draft-ietf-magma-mld-source-03.txt
    - draft-ietf-magma-mld-source-02.txt

- RFCs:
  - RFC 3228: IANA Considerations for IGMP

Back

MALLOC - Multicast Address Allocation (MALLOC@IETF)

- Drafts:
  - Multicast Address Allocation MIB (ietf/draft-ietf-malloc-malloc-mib-07.txt)
- RFCs:
  - RFC 2730: Multicast Address Dynamic Client Allocation Protocol (MADCAP)
  - RFC 2771: An Abstract API for Multicast Address Allocation
  - RFC 2907: MADCAP Multicast Scope Nesting State Option
  - RFC 2908: The Internet Multicast Address Allocation Architecture
  - RFC 2909: The Multicast Address Set Claim (MASC) Protocol
  - RFC 3307: Dynamic Allocation Guidelines for IPv6 Multicast Addresses

Back

MBONED - MBone Deployment WG (MBONED@IETF)

- Drafts:
  - Anycast RP mechanism using PIM and MSDP (draft-ietf-mboned-anycast-rp-08.txt)
  - IPv4 Automatic Multicast Without Explicit Tunnels (draft-ietf-mboned-auto-multicast-01.txt)
  - IANA Guidelines for IPv4 Multicast Address Assignments (draft-ietf-mboned-rfc3171-update-00.txt)
  - Unicast-Prefix-based IPv4 Multicast Addresses (draft-ietf-mboned-ipv4-uni-based-mcast-00.txt)
  - Internet Multicast Gap Analysis from the MBONED Working Group for the IESG (draft-ietf-mboned-iesg-gap-analysis-00.txt)
  - Source-Specific Protocol Independent Multicast in 232/8 (draft-ietf-mboned-ssm232-04.txt)
- RFCs:
  - RFC 2365: Administratively Scoped IP Multicast
  - RFC 2588: IP Multicast and Firewalls
  - RFC 2770: GLOP Addressing in 233/8
  - RFC 2776: Multicast-Scope Zone Announcement Protocol (MZAP)
  - RFC 3138: Extended Allocations in 233/8
  - RFC 3171: IANA Guidelines for IPv4 Multicast Address Assignments
  - RFC 3170: IP Multicast Applications: Challenges and Solutions
  - RFC 3180: GLOP Addressing in 233/8

Back

MOSPF - Multicast Extensions to OSPF (MOSPF@IETF)

- Drafts:
  - <none>
- RFCs:
  - RFC 1469: IP Multicast over Token-Ring Local Area Networks
  - RFC 1584: Multicast Extensions to OSPF
  - RFC 1585: MOSPF: Analysis and Experience

Back

MSDP - Multicast Source Discovery Protocol (MSDP@IETF)

- Drafts:
  - Multicast Source Discovery Protocol (MSDP)
    - draft-ietf-msdp-spec-14.txt
    - draft-ietf-msdp-spec-08.txt
- RFCs:
  - <none>

Back

MSEC - Multicast Security (MSEC@IETF)

- Drafts:
  - The Group Domain of Interpretation (draft-ietf-msec-gdoi-06.txt)
  - Group Key Management Architecture (draft-ietf-msec-gkmarch-03.txt)
  - GSAKMP Light (draft-ietf-msec-gsakmp-light-sec-01.txt)
  - MIKEY: Multimedia Internet KEYing (draft-ietf-msec-mikey-04.txt)
  - HMAC-authenticated Diffie-Hellman for MIKEY (draft-ietf-msec-mikey-dhhmac-00.txt)
- RFCs:
  - <none>

PIM - Protocol Independent Multicast (PIM@IETF)

- Drafts:
  - Bi-directional Protocol Independent Multicast (BIDIR-PIM) (draft-ietf-pim-bidir-04.txt)
  - Protocol Independent Multicast MIB
    - draft-ietf-pim-mib-v2-01.txt
    - draft-ietf-pim-mib-v2-00.txt
  - Protocol Independent Multicast - Sparse Mode (PIM-SM): Protocol Specification
    - draft-ietf-pim-sm-v2-new-06.txt
    - draft-ietf-pim-sm-v2-new-04.txt
  - Bootstrap Router (BSR) Mechanism for PIM Sparse Mode (draft-ietf-pim-sm-bsr-02.txt)
  - Protocol Independent Multicast - Dense Mode (PIM-DM)Protocol Specification (Revised)
    - draft-ietf-pim-dm-new-v2-03.txt
    - draft-ietf-pim-dm-new-v2-02.txt
- RFCs:
  - <none>

Back

RMT - Reliable Multicast Transport (RMT@IETF)

- Drafts:
  - TRACK PROTOCOL INSTANTION OVER UDP:
    TREE ACKNWOLEGEMENT BASED RELIABLE MULTICAST
    (draft-whetten-rmt-track-pi-00.txt)
  - NACK-Oriented Reliable Multicast Protocol (NORM) (draft-ietf-rmt-pi-norm-05.txt)
  - Forward Error Correction building block (draft-ietf-rmt-bb-fec-07.txt)
  - Asynchronous Layered Coding protocol instantiation (draft-ietf-rmt-pi-alc-08.txt)
  - NACK-Oriented Reliable Multicast (NORM) Protocol Building Blocks
    (draft-ietf-rmt-bb-norm-04.txt)
  - Layered Coding Transport transport building block (draft-ietf-rmt-bb-lct-04.txt)
  - The use of Forward Error Correction in Reliable Multicast (draft-ietf-rmt-info-fec-03.txt)
  - PGMCC single rate multicast congestion control: Protocol Specification

(draft-ietf-rmt-bb-pgmcc-01.txt)
    - Wave and Equation Based Rate Control building block
        - draft-ietf-rmt-bb-webrc-04.txt
        - draft-ietf-rmt-bb-webrc-03.txt
        - draft-ietf-rmt-bb-webrc-02.txt
    - Reliable Multicast Transport Building Block Generic Router Assist - Signalling Protocol Specification (draft-ietf-rmt-bb-gra-signalling-00.txt)
    - TCP-Friendly Multicast Congestion Control (TFMCC):Protocol Specification (draft-ietf-rmt-bb-tfmcc-01.txt)
    - Reliable Multicast Transport Building Block for TRACK (draft-ietf-rmt-bb-track-02.txt)
    - Compact Forward Error Correction (FEC) Schemes (draft-ietf-rmt-bb-fec-supp-inband-00.txt)
- RFCs:
    - RFC 2887: The Reliable Multicast Design Space for Bulk Data Transfer
    - RFC 3048: Reliable Multicast Transport Building Blocks for One-to-Many Bulk-Data Transfer
    - RFC 3208: PGM
    - RFC 3269: Author Guidelines for RMT Building Blocks and Protocol Instantiation documents

Back

SSM - Source Specific Multicast WG (SSM@IETF)

- Drafts:
    - Source-Specific Multicast for IP (draft-ietf-ssm-arch-01.txt)
    - An Overview of Source-Specific Multicast(SSM) Deployment (draft-ietf-ssm-overview-04.txt)
- RFCs:
    - <none>

Back

IETF - Related Work (Drafts, RFCs)

- Drafts:
    - Connectionless Multicast (draft-ooms-cl-multicast-02.txt)
    - Extended RSVP-TE for Multicast LSP Tunnels (draft-yasukawa-mpls-rsvp-multicast-00.txt)
    - MPLS for PIM-SM (draft-ooms-mpls-pimsm-00.txt)
    - MPLS Multicast Traffic Engineering (draft-ooms-mpls-multicast-te-01.txt)
    - Multicast in MPLS/BGP VPNs (draft-rosen-vpn-mcast-04.txt)
    - Overview of Multicast in VPNs (draft-ooms-ppvpn-mcast-overview-00.txt)
- RFCs:
    - MPLS WG: RFC 3353: Framework for IP Multicast in MPLS

Title : Multicast Control Protocol (MCOP)
Author(s) : R. Lehtonen et al.
Filename : draft-lehtonen-magma-mcop-02.txt

Pages : 39
Date : 2003-2-12

This draft introduces Multicast Control Protocol (MCOP) that may be used as a tool for multicast network management. MCOP provides multicast network remote management with centralized information database located at Multicast Control Server (MCS). It allows gradual, group and network specific multicast network deployment. MCOP protocol is used between MCS and routers that have directly connected multicast sources or receivers. The actual control is done by MCOP enabled routers based on the information received from the MCS. MCOP router can filter IGMP/MLD reports and multicast packets before they reach the IGMP/MLD processing layer or multicast routing stack of the router. MCOP is independent of multicast routing protocols.

Title : An Effective Solution for Multicast Scalability:The MPLS Multicast Tree (MMT)
Author(s) : A. Boudani, B. Cousin, J. Bonnin
Filename : draft-boudani-mpls-multicast-tree-03.txt
Pages : 10
Date : 2003-2-10

A multicast router should keep forwarding state for every multicast tree passing through it. The number of forwarding states grows with the number of groups. In this paper, we present a new approach, the MPLS multicast Tree (MMT), which utilizes MPLS LSPs between multicast tree branching node routers in order to reduce forwarding states and enhance scalability. In our approach only routers that are acting as multicast tree branching node routers for a group need to keep forwarding state for that group. All other non-branching node routers simply forward data packets by traffic engineered unicast routing using MPLS LSPs. We can deduce that our approach can be largely deployed because it uses for multicast traffic the same unicast MPLS forwarding scheme. We will briefly discuss the multicast scalability problem, related works and different techniques for forwarding state reduction. We discuss also the advantages of our approach, and conclude that it is feasible and promising. Finally, we analytically evaluate our approach.

Title : Framework of Overlay Multicast Control Protocol
Author(s) : S. Koh, J. Park
Filename : draft-sjkoh-overlay-multicast-framework-00.txt
Pages : 15
Date : 2003-2-7

This document describes Overlay Multicast Control Protocol (OMCP) used for realizing and managing Overlay Multicast, as also known as Application Layer Multicast. Overlay Multicast is a data delivery scheme for multicast applications, in which one or more intermediate relay agents are employed for relaying application data from a sender to many receivers along a pre-configured or automatically configured tree hierarchy. For standardization of Overlay Multicast, it needs to separate the data plane (for packet relaying) and the control plane (for tree configuration and session monitoring). We describe a control protocol for Overlay

Multicast, named Overlay Multicast Control Protocol (OMCP). In OMCP, a special-purpose entity, called Session Manager, is used to manage and control the tree configuration and session monitoring. OMCP is designed to ensure that the multicast applications and services can be provided over current Internet environments in which IP multicast has not completely been deployed.

Title : Duplicate Address Detection Optimization using IPv6 Multicast Listener Discovery
Author(s) : G. Daley, R. Nelson
Filename : draft-daley-ipv6-mcast-dad-02.txt
Pages : 12
Date : 2003-2-3

This draft describes a possible optimization to Duplicate Address Detection (DAD) which can be used to successfully terminate DAD early, based on the presence of listeners on the link-scope solicited nodes multicast address.

Title : Anycast-RP using PIM
Author(s) : D. Farinacci, Y. Cai
Filename : draft-farinacci-pim-anycast-rp-00.txt
Pages : 7
Date : 2003-1-22

This proposal allows Anycast-RP to be used inside a domain which runs PIM only. There are no other multicast protocols required to support Anycast-RP, such as MSDP, which has been used traditionally to solve this problem.

Title : Explicit Multicast over Mobile IP (XMIP)
Author(s) : J. Lee
Filename : draft-lee-xcast-mobileip-01.txt
Pages : 17
Date : 2003-1-17

As a special kind of Internet multicast, Explicit multicast(Xcast)[1] encodes destination addresses of all the receivers into the packet. Not requiring membership management and routing information exchange in the intermediate routers in contrast to the legacy Internet multicast or Deering multicast, Xcast can effectively provide multicast service to Internet without those overheads. A node mobility supporting protocol, Mobile IP[2,3], requires to be modified to appropriately intercept, route and forward Xcast packets. This document specifies the protocol operations for the mobile agent, the mobile node and the correspondent node to support transmission and reception of Xcast datagram over Mobile IPv4 and Mobile IPv6.

Title : Explicit Multicast (Xcast) Basic Specification

Author(s) : R. Boivie et al.
Filename : draft-ooms-xcast-basic-spec-04.txt
Pages : 29
Date : 2003-1-17

Multicast has become increasingly important with the emergence of network-based applications such as IP telephony and video conferencing. The Internet community has done a significant amount of work on IP multicast over the last decade [1075, 2201, 2236, DEER, DEE2, FARI, HOLB, MBONE, PERL]. However, while today's multicast schemes are scalable in the sense that they can support very large multicast groups, there are scalability issues when a network needs to support a very large number of distinct multicast groups.

Title : Embedding the Address of RP in IPv6 Multicast Address
Author(s) : P. Savola, B. Haberman
Filename : draft-savola-mboned-mcast-rpaddr-01.txt
Pages : 11
Date : 2003-2-4

As has been noticed, there is exists a huge deployment problem with global, interdomain IPv6 multicast: PIM RPs have no way of communicating the information about multicast sources to other multicast domains, as there is no MSDP, and the whole interdomain Any Source Multicast model is rendered unusable; SSM avoids these problems. This memo outlines a way to embed the address of the RP in the multicast address, solving the interdomain multicast problem. The problem is three-fold: specify an address format, adjust the operational procedures and configuration if necessary, and modify PIM implementations of those who want to join a group (DR's) or create one (RP's). In consequence, there would be no need for interdomain MSDP.

Title : Performing DNS queries via IP Multicast
Author(s) : S. Cheshire
Filename : draft-cheshire-dnsext-multicastdns-01.txt
Pages : 29
Date : 2002-12-23

As networked devices become smaller, more portable, and more ubiquitous, the ability to operate with less configured infrastructure is increasingly important. In particular, the ability to look up host names and similar DNS resource record data types, in the absence of a conventional managed DNS server, is becoming essential.

Title : Multicast Listener Discovery Version 2 (MLDv2) for IPv6
Author(s) : R. Vida, L. Costa
Filename : draft-vida-mld-v2-06.txt
Pages : 50
Date : 2002-12-2

This document specifies Version 2 of the Multicast Listener Discovery Protocol, MLDv2. MLD is the protocol used by an IPv6 router to discover the presence of multicast listeners (that is, nodes wishing
to receive multicast packets) on its directly attached links, and to discover specifically which multicast addresses are of interest to those neighboring nodes. MLDv2 is derived from version 3 of IPv4's Internet Group Management Protocol, IGMPv3. Compared to the previous version, MLDv2 adds
support for 'source filtering', that is, the ability for a node to report interest in listening to packets *only* from specific source addresses, or from *all but* specific source addresses, sent to a particular multicast address. This document obsoletes RFC 2710.

Title : MultiGRIP: Quality of Service Aware Multicasting over DiffServ
Author(s) : G. Bianchi et al.
Filename : draft-bianchi-qos-multicast-over-diffserv-00.txt
Pages : 19
Date : 2002-12-2

Efficient delivery of real-time multicast traffic imposes on the underlying network infrastructure the burden of supporting Quality of Service (QoS). This can be quite a complex issue in a Differentiated Services (DiffServ) IP network, especially if multicast users are allowed to dynamically join and leave the multicast tree. In fact, since DiffServ lacks of explicit reservation states, i) a replicating node cannot test whether a corresponding reservation exists on an output link, and ii) upon a dynamic join of a QoS multicast user, the DiffServ network lacks of control functions to verify whether resources are available along the new path. In this document, we present a solution to support dynamic multicast with QoS over a DiffServ network. Our solution combines two ideas. First, resource availability along a new QoS path is verified via a probe-based approach. Second, QoS is maintained by marking replicated packets with a special DSCP value, before forwarding them on the QoS path. We are fully aware that the possible application of the principles described in this draft in the Internet raises many issues, which we do not address. Our aim, then, is not proposing a fully-fledged solution, but contributing to the on-going discussions in the international arena on these matters, by means of what we may see also as a problem statement document.

Title : Methodology for IP Multicast Benchmarking
Author(s) : H. Soor, D. Stopp
Filename : draft-ietf-bmwg-mcastm-10.txt
Pages : 22
Date : 2002-12-2

The purpose of this draft is to describe methodology specific to the benchmarking of multicast IP forwarding devices. It builds upon the tenets set forth in RFC 2544, RFC 2432 and other IETF Benchmarking Methodology Working Group (BMWG) efforts. This document seeks to extend these efforts to the multicast paradigm. The BMWG produces two major classes of documents: Benchmarking Terminology documents and Benchmarking Methodology documents. The Terminology documents present the benchmarks and other related terms. The

Methodology documents define the procedures required to collect the benchmarks cited in the corresponding Terminology documents.

Title : IP Multicast in Differentiated Services Networks
Author(s) : R. Bless, K. Wehrle
Filename : draft-bless-diffserv-multicast-05.txt
Pages : 33
Date : 2002-11-22

This document identifies problems which will arise when IP Multicast is used in Differentiated Services (DS) networks. Although the basic DS forwarding mechanisms also work with IP Multicast, some facts have to be considered which are related to the provisioning of multicast resources. The presented problems mainly lead to situations in which other service users are affected adversely in their experienced quality. An adequate solution is described in this document. It provides the necessary resource decoupling for protecting reserved resources until admission control is performed.

Title : RTCP Extensions for Single-Source Multicast Sessions with Unicast Feedback
Author(s) : J. Chesterfield et al.
Filename : draft-ietf-avt-rtcpssm-02.txt
Pages : 28
Date : 2002-11-11

This document specifies a modification to the Real-time Transport Control Protocol (RTCP) to use unicast feedback. The proposed extension is useful for single source multicast sessions such as Source Specific Multicast (SSM) communication where the traditional model of many-to-many group communication is either not possible or not preferred. In addition, it can be applied to any group that might benefit from a sender controlled summarised reporting mechanism.

Title : On-Demand Multicast Routing Protocol (ODMRP) for Ad-Hoc Networks
Author(s) : Y. Yi, S. Lee
Filename : draft-ietf-manet-odmrp-04.txt
Pages : 29
Date : 2002-11-7

On-Demand Multicast Routing Protocol (ODMRP) is a multicast routing protocol designed for ad-hoc networks with mobile hosts. ODMRP is a mesh-based, rather than a conventional tree-based, multicast scheme and uses a Forwarding Group concept (only a subset of nodes forwards the multicast packets via scoped flooding). It applies on-demand procedures to dynamically set up routes and maintain multicast group membership. ODMRP is well suited for ad-hoc wireless networks with mobile hosts where bandwidth is limited, topology changes frequently and rapidly, and power is constrained.

Title : TCP-Friendly Multicast Congestion Control (TFMCC):Protocol Specification
Author(s) : J. Widmer, M. Handley
Filename : draft-ietf-rmt-bb-tfmcc-01.txt
Pages : 29
Date : 2002-11-4

This document specifies TCP-Friendly Multicast Congestion Control (TFMCC). TFMCC is a congestion control mechanism for multicast transmissions in a best-effort Internet environment. It is a single-rate congestion control scheme, where the sending rate is adapted to the receiver experiencing the worst network conditions. TFMCC is reasonably fair when competing for bandwidth with TCP flows and has a relatively low variation of throughput over time, making it suitable for applications such as streaming media where a relatively smooth

---

Title : Considerations for IGMP and MLD snooping switches
Author(s) : M. Christensen, K. Kimball
Filename : draft-ietf-magma-snoop-03.txt
Pages : 11
Date : 2002-11-4

This memo describes the requirements for IGMP and MLD snooping switches. The requirements for IGMPv2 snooping switches are based on best current practices. IGMPv3 and MLDv2 snooping are also covered in this draft although we are not aware of any such implementations at the time of writing. Note that IGMP snooping is related only to IPv4 multicast. Other multicast packets, such as IPv6, might be suppressed by the snooping functionality if additional care is not taken in the implementation. It is desired not to restrict the flow of non-IPv4 multicasts other than to the degree which would happen as a result of regular bridging functions. The same note can be made of MLD snooping switches with respect to suppression of IPv4.

Title : IPv6 Multicast Deployment Issues
Author(s) : P. Savola
Filename : draft-savola-v6ops-multicast-issues-01.txt
Pages : 8
Date : 2002-11-4

There are many issues concerning the deployment and implementation, and to a lesser degree, specification of IPv6 multicast. This memo describes known problems, trying to raise awareness. Currently, global IPv6 interdomain multicast is completely impossible except using SSM: there is no way to convey information about multicast sources between PIM RPs. Site-scoped multicast is also problematic when used alongside to global multicast because of that. A few possible solutions are outlined or referred to. In addition, an issue regarding link-local multicast-blocking Ethernet switches is brought up. Finally, a feature request for MLD snooping switches is noted.

Title : Extended RSVP-TE for Multicast LSP Tunnels
Author(s) : S. Yasukawa et al.
Filename : draft-yasukawa-mpls-rsvp-multicast-01.txt
Pages : 46
Date : 2002-11-4

Multicast technology will become increasingly important with the dissemination of new applications such as contents delivery services and video conferences, which require much more bandwidth and stricter QoS than conventional applications. From the service providers' perspective, traffic engineering (TE) functions will be needed to handle the large amount of multicast traffic.This document defines some protocol extensions to the existing RSVP-TE[1] in order to establish a multicast label switched path (LSP). The use of label switching routers (LSRs) with these protocol extensions defined in this document allows service providers to offer unicast and multicast multiprotocol label switching (MPLS) services in the same service network. This protocol assumes a variable LSP topology, e.g., point-to-multipoint, multipoint-to-multipoint, topologies. This document describes how to establish point-to-multipoint and multipoint-to-multipoint LSPs as the most basic multicast topology. It defines two ways of constructing a point-to-multipoint LSP: sender-initiated LSP setup and leaf-initiated LSP setup. Each method has an LSP modification function in order to adapt to dynamic changes in the LSP tree topology. This MPLS architecture[10] is very flexible and can be expanded to carry protocols other than IP multicasting, e.g., Ethernet, PPP, and SONET/SDH, but this document only defines IP multicasting (IPv4 and IPv6) as a forwarding equivalence class object (FEC).

Title : Cisco Systems Router-port Group Management Protocol (RGMP)
Author(s) : I. Wu, T. Eckert
Filename : draft-wu-rgmp-03.txt
Pages : 16
Date : 2002-11-4

This draft documents RGMP, a protocol developed by Cisco Systems that is used between multicast routers and switches to restrict multicast packet forwarding in switches to those routers where the packets may be needed. RGMP is designed for backbone switched networks where multiple, high speed routers are interconnected.

Title : An Architecture of Overlay Multicast Control Protocol
Author(s) : S. Koh et al.
Filename : draft-koh-omcp-00.txt
Pages : 16
Date : 2002-11-4

This document describes Overlay Multicast Control Protocol (OMCP) used for realizing and managing Overlay Multicast, as also known as Application Layer Multicast. Overlay Multicast is a data delivery scheme for multicast applications, in which one or more intermediate relay

agents are employed for relaying application data from a sender to many receivers along a pre-configured or automatically configured tree hierarchy. In OMCP, a special- purpose entity, called Session Manager, is used to manage and control the tree configuration and session monitoring. OMCP is designed to ensure that the multicast applications and services can be provided over current Internet environments in which IP multicast has not completely been deployed.

Title : Protocol Independent Multicast MIB
Author(s) : J. Nicholas et al.
Filename : draft-ietf-pim-mib-v2-01.txt
Pages : 31
Date : 2002-11-1

This memo defines a portion of the Management Information Base (MIB) for use with network management protocols in the Internet community. In particular, it describes managed objects used for managing the Protocol Independent Multicast (PIM) protocol.

Title : Protocol Independent Multicast - Dense Mode (PIM-DM)Protocol Specification (Revised)
Author(s) : A. Adams, J. Nicholas, W. Siadak
Filename : draft-ietf-pim-dm-new-v2-02.txt
Pages : 54
Date : 2002-11-1

This document specifies Protocol Independent Multicast - Dense Mode (PIM-DM). PIM-DM is a multicast routing protocol that uses the underlying unicast routing information base to flood multicast datagrams to all multicast routers. Prune messages are used to prevent future messages from propagating to routers with no group membership information.

Title : Source Address Selection for Multicast Listener Discovery Protocol (RFC 2710)
Author(s) : B. Haberman
Filename : draft-ietf-magma-mld-source-03.txt
Pages : 4
Date : 2002-11-1

It has come to light that there is an issue with the selection of a suitable IPv6 source address for Multicast Listener Discovery messages when a node is performing stateless address autoconfiguration. This memo is intended to clarify the rules on selecting an IPv6 address to use for MLD messages.

Title : VPLS based on IP Multicast
Author(s) : A. Sajassi, H. Salama
Filename : draft-sajassi-mvpls-00.txt

Pages : 14
Date : 2002-11-1

Virtual Private LAN Service (VPLS) is a type of Layer-2 Provider Provisioned Virtual Private Network (L2 PPVPN) offered service, which has been described in [PPVPN-FRWK]. If the Service Provider network provides IP multicast functionality, then this network capability can be leveraged in providing very efficient VPLS service and yet simplifying the implementation of such service. This document describes a solution for providing VPLS service based on IP multicast feature referred to as Multicast Virtual Private LAN Service (MVPLS).

Title : Xcast+ Extension for Few-to-Few Mulitcast Communication
Author(s) : K. Kim et al.
Filename : draft-kim-xcast+-few-2-few-00.txt
Pages : 14
Date : 2002-10-31

Xcast+[2] is newly proposed multicast scheme to support IGMPv2/MLDv3 in Explicit Multicast(Xcast)[1]. Xcast+ is suitable for one-to-few multicast applications. In order to support few-to-few communication naturally, the control plane of existing Xcast+ should be extended. In this document, a new control plane for few-to-few communication for Xcast+ is specified. In order to acheive this, the concept of logical core (LC) is newly introduced.

Title : IGMP for user Authentication Protocol (IGAP)
Author(s) : T. Hayashi et al.
Filename : draft-hayashi-igap-00.txt
Pages : 22
Date : 2002-10-30

IP Multicast applications are becoming more common. Two key concerns raised by the providers of such applications are the lack of control on what users can get multicast traffic and a method for tracking user usage (such as how long these users are joined to a multicast group). This document introduces the IGMP for user Authentication Protocol (IGAP). IGAP extends the existing IGMPv2 protocol to add user authentication functionality. IGAP enables an IP multicast service provider to authenticate requests to join a specific multicast group based on user information.

Title : Selectively Reliable Multicast Protocol (SRMP)
Author(s) : M. Pullen et al.
Filename : draft-pullen-srmp-00.txt
Pages : 26
Date : 2002-10-29

The Selectively Reliable Multicast Protocol (SRMP) is a transport protocol, intended to deliver a mix of reliable and best-effort messages in an any-to-any multicast environment, where the

best-effort traffic occurs in significantly greater volume than the best-effort traffic and therefore can be used to carry sequence numbers of reliable messages for loss detection. SRMP is intended for use in a distributed simulation application environment, where only the latest value of reliable transmission for any particular data identifier requires delivery. SRMP has two sublayers: a bundling sublayer that combines short messages, peforms NAK suppression, and incorporates the TCP-Friendly Multicast Congestion Control of Widmer and Handley, dropping best-effort traffic in order to achieve congestion control; and a selectively reliable transport (SRT) sublayer that formats best-effort and reliable transmissions and also creates negative acknowledgements when loss of reliable messages is detected.

Title : IP Version 6 Addressing Architecture
Author(s) : B. Hinden, S. Deering
Filename : draft-ietf-ipngwg-addr-arch-v3-11.txt
Pages : 26
Date : 2002-10-28

This specification defines the addressing architecture of the IP Version 6 protocol [IPV6]. The document includes the IPv6 addressing model, text representations of IPv6 addresses, definition of IPv6 unicast addresses, anycast addresses, and multicast addresses, and an IPv6 node's required addresses. This document obsoletes RFC-2373 'IP Version 6 Addressing Architecture'.

Title : SDP Source-Filters
Author(s) : B. Quinn, R. Finlayson
Filename : draft-ietf-mmusic-sdp-srcfilter-02.txt
Pages : 10
Date : 2002-10-25

This document describes how to adapt the Session Description Protocol (SDP) to express one or more source addresses as a source filter for one or more destination 'connection' addresses. It defines the syntax and semantics for an SDP 'source-filter' attribute that may reference either IPv4 or IPv6 address(es) as either an inclusive or exclusive source list for either multicast or unicast destinations. In particular, an inclusive source-filter can be used to specify a Source-Specific Multicast ('SSM') session.

Receiver applications are expected use the SDP source-filter information to identify traffic from legitimate senders and discard traffic from illegitimate senders. Applications and hosts may also share the source-filter information with network elements (e.g., with routers using IGMPv3) so they can potentially perform the traffic filtering operation further 'upstream,' closer to the source(s).

Title : Analysis on RSVP Regarding Multicast
Author(s) : X. Fu, C. Kappler, H. Tschofenig
Filename : draft-fu-rsvp-multicast-analysis-01.txt

Pages : 14
Date : 2002-10-25

RSVP version 1 has been designed for optimum support multicast.However, in reality multicast is being used much less frequently than anticipated. Still, even for unicast (one sender, one receiver) full-fledged multicast-enabled RSVP signaling must be used. As pointed out in the NSIS requirement draft, multicast would not be necessarily required for an NSIS signaling protocol. This draft analyses ingredients of RSVP Version 1 which are affected by multicast, and derives how these ingredients may look like if multicast is not supported in the generic RSVP signaling protocol and adapt related functionalities accordingly - we call the resulting feature set 'RSVP Lite', a potentially more light-weight version of RSVP.

Title : Methodology for IP Multicast Benchmarking
Author(s) : H. Soor, D. Stopp
Filename : draft-ietf-bmwg-mcastm-09.txt
Pages : 21
Date : 2002-10-24

The purpose of this draft is to describe methodology specific to the benchmarking of multicast IP forwarding devices. It builds upon the tenets set forth in RFC 2544, RFC 2432 and other IETF Benchmarking Methodology Working Group (BMWG) efforts. This document seeks to extend these efforts to the multicast paradigm. The BMWG produces two major classes of documents: Benchmarking Terminology documents and Benchmarking Methodology documents. The Terminology documents present the benchmarks and other related terms. The Methodology documents define the procedures required to collect the benchmarks cited in the corresponding Terminology documents.

Title : Simple Explicit Multicast (SEM)
Author(s) : A. Boudani, B. Cousin
Filename : draft-boudani-simple-xcast-02.txt
Pages : 12
Date : 2002-10-23

In this document, we propose a new multicast protocol called SEM (Simple Explicit Multicast) or simple Xcast[1]. This protocol uses an efficient method to construct the multicast tree and deliver multicast packets. In order to construct the multicast tree, this protocol uses the same mecanism as Xcast+[2].For the delivery of multicast packets it uses the mechanism of branching routers similar to the mechanism used in REUNITE I [3] and REUNITE II[4].

Title : Multicast Control Protocol (MCOP)
Author(s) : R. Lehtonen et al.
Filename : draft-lehtonen-magma-mcop-01.txt
Pages : 34
Date : 2002-10-18

In IP multicast all hosts that join a multicast group (*, G) or (S,G) can receive the multicast traffic. This draft introduces Multicast Control Protocol (MCOP) that makes it possible to selectively enable multicast receiving and sending. MCOP is used between Multicast Control Agent (MCA) and routers that have directly connected multicast sources or receivers. The receiver and source control is done by MCOP enabled routers based on the information received from the MCA. MCOP enabled routers filter IGMP/MLD reports and multicast packets before they reach the IGMP/MLD processing layer or multicast routing stack of the router. MCOP is independent of multicast routing protocols.

Title : Multicast Listener Discovery Version 2 (MLDv2) for IPv6
Author(s) : R. Vida et al.
Filename : draft-vida-mld-v2-05.txt
Pages : 52
Date : 2002-10-16

This document specifies Version 2 of the Multicast Listener Discovery protocol, MLDv2. MLD is the protocol used by an IPv6 router to discover the presence of multicast listeners (that is, nodes wishing to receive multicast packets) on its directly attached links, and to discover specifically which multicast addresses are of interest to those neighboring nodes. MLDv2 is derived from version 3 of IPv4's Internet Group ManagementProtocol, IGMPv3. Compared to the previous version, MLDv2 adds support for 'source filtering', that is, the ability for a node to report interest in listening to packets *only* from specific source addresses, or from *all but* specific source addresses, sent to a particular multicast address. That information may be used by multicast routing protocols to avoid delivering multicast packets from specific sources to links where there are no interested listeners. When compared to IGMPv3, one important difference to note is that MLDv2 uses ICMPv6 (IP Protocol 58) message types, rather than IGMP (IP Protocol 2) message types.

Title : An IPv6/IPv4 Multicast Translator based on IGMP/MLD Proxying (mtp)
Author(s) : K. Tsuchiya, H. Higuchi, S. Sawada, S. Nozaki
Filename : draft-ietf-ngtrans-mtp-03.txt
Pages : 14
Date : 2002-10-15

In the stage of the transition from IPv4 to IPv6 it is necessary that IPv4 nodes and IPv6 nodes can communicate directly. This memo proposes a mechanism which enables such direct communication for multicast, in addition to that for unicast defined in [SIIT] and [NAT-PT].

Title : Embedding the Address of RP in IPv6 Multicast Address
Author(s) : P. Savola, B. Haberman
Filename : draft-savola-mboned-mcast-rpaddr-00.txt
Pages : 10
Date : 2002-10-10

As has been noticed, there is exists a huge deployment problem with global, interdomain IPv6 multicast: PIM RPs have no way of communicating the information about multicast sources to other multicast domains, as there is no MSDP, and the whole interdomain Any Source Multicast model is rendered unusable; SSM avoids these problems. This memo outlines a way to embed the address of the RP in the multicast address, solving the interdomain multicast problem. The problem is three-fold: specify an address format, adjust the operational procedures and configuration if necessary, and modify receiver-side PIM implementations. In consequence, there would be no need for interdomain MSDP.

Title : Duplicate Address Detection Optimization using IPv6 Multicast Listener Discovery
Author(s) : G. Daley, R. Nelson
Filename : draft-daley-ipv6-mcast-dad-01.txt
Pages : 12
Date : 2002-10-8

This draft describes a possible optimization to Duplicate Address Detection (DAD) which can be used to successfully terminate DAD early, based on the presence of listeners on the link-scope solicited nodes multicast address.

Title : IPv6 Multicast Deployment Issues
Author(s) : P. Savola
Filename : draft-savola-v6ops-multicast-issues-00.txt
Pages : 7
Date : 2002-10-7

There are many issues concerning the deployment and implementation, and to a lesser degree, specification of IPv6 multicast. This memo describes known problems, trying to raise awareness. Currently, global IPv6 interdomain multicast is completely impossible except using SSM: there is no way to convey information about multicast sources between PIM RPs. Site-scoped multicast is also problematic when used alongside to global multicast because of that. A few possible solutions are outlined or referred to. In addition, an issue regarding link-local multicast-blocking Ethernet switches is brought up. Finally, a feature request for MLD snooping switches is noted.

Title : M-ISIS: Multi Topology (MT)Routing in IS-IS
Author(s) : T. Przygienda, N. Shen, N. Sheth
Filename : draft-ietf-isis-wg-multi-topology-05.txt
Pages : 11
Date : 2002-10-2

This draft describes an optional mechanism within ISIS used today by many ISPs for IGP routing within their clouds. This draft describes how to run within a single ISIS domain a set of independent IP topologies that we call Multi-Topologies (MTs). This MT extension can be

used for variety of purposes such as an in-band management network ``on top'' of the original IGP topology, maintain separate IGP routing domains for isolated multicast or IPv6 islands within the backbone, or force a subset of an address space to follow a different topology.

Title : A MAPOS NSP (Node Switch Protocol) Multicast Expansion - NSP+
Author(s) : T. Ogura et al.
Filename : draft-ogura-mapos-nsp-multiexp-00.txt
Pages : 13
Date : 2002-9-27

This document describes NSP+, an expansion of the MAPOS NSP (Node Switch Protocol). MAPOS is a multiple access protocol for transmission of network-protocol datagrams, encapsulated in High-Level Data Link Control (HDLC) frames, over SONET/SDH. NSP is a protocol for automatically assigning MAPOS node addresses. Other than the same function as NSP, NSP+ has an additional function to reduce unnecessary multicast frame transmission on MAPOS switches. In NSP+, a node can send a list of multicast HDLC addresses to a directly connected switch to notify that, in the case of multicasting, the code needs to receive only those frames whose destination addresses are included in the list. This enables the switch to forward only required multicast frames to directly connected nodes and reduce unnecessary bandwidth consumption.

Title : Requirements for Automatic Configuration of IP Hosts
Author(s) : A. Williams
Filename : draft-ietf-zeroconf-reqts-12.txt
Pages : 22
Date : 2002-9-25M

Many common TCP/IP protocols such as DHCP [RFC2131], DNS [RFC1034][RFC1035], MADCAP [RFC2730], and LDAP [RFC2251] must be configured and maintained by an administrative staff. This is unacceptable for emerging networks such as home networks, automobile networks, airplane networks, or ad hoc networks at conferences, emergency relief stations, and many others. Such networks may be nothing more than two isolated laptop PCs connected via a wireless LAN. For all these networks, an administrative staff will not exist and the users of these networks neither have the time nor inclination to learn network administration skills. Instead, these networks need protocols that require zero user configuration and administration.This document is part of an effort to define such zero configuration (zeroconf) protocols. Before embarking on defining zeroconf protocols, protocol requirements are needed. This document states the zeroconf protocol requirements for four protocol areas; they are: IP interface configuration, translation between host name and IP address, IP multicast address allocation, and service discovery. This document does not define specific protocols, just requirements. The requirements for these four areas result from examining everyday use or scenarios of these protocols.

Title : The UDP Multicast Tunneling Protocol

Author(s) : R. Finlayson
Filename : draft-finlayson-umtp-07.txt
Pages : 6
Date : 2002-9-23

Many Internet hosts - such as PCs - while capable of running multicast applications, cannot access the MBone (or other wide-area multicast network) because the router(s) that connect them to the Internet do not yet support IP multicast routing. The 'UDP Multicast Tunneling Protocol' (UMTP) enables such a host to establish an 'ad hoc' connection to the MBone by tunneling multicast UDP datagrams inside unicast UDP datagrams. By using UDP, this tunneling can be implemented as a 'user level' application, without requiring changes to the host's operating system.

Title : Definitions of Managed Objects for Bridges with Traffic Classes, Multicast Filtering and Virtual LAN
Extensions
Author(s) : V. Ngai, E. Bell et al.
Filename : draft-ietf-bridge-ext-v2-01.txt
Pages : 88
Date : 2002-9-23

This memo defines a portion of the Management Information Base (MIB) for use with network management protocols in TCP/IP based internets. In particular, it defines two MIB modules for managing the new capabilities of MAC bridges defined by the IEEE 802.1D-1998 MAC Bridges and the IEEE 802.1Q-1998 Virtual LAN (VLAN) standards for bridging between Local Area Network (LAN) segments. One MIB module defines objects for managing the 'Traffic Classes' and 'Enhanced Multicast Filtering' components of IEEE 802.1D-1998 and P802.1t-2001. The other MIB module defines objects for managing VLANs, as specified in IEEE 802.1Q-1998, P802.1u and P802.1v. Provisions are made for support of transparent bridging. Provisions are also made so that these objects apply to bridges connected by subnetworks other than LAN segments. This memo also includes several MIB modules in a manner that is compliant to the SMIv2 [V2SMI]. This memo supplements RFC 1493 [BRIDGEMIB] and (to a lesser extent) RFC 1525 [SBRIDGEMIB].

Title : The MIDI Wire Protocol Packetization (MWPP)
Author(s) : J. Lazzaro, J. Wawrzynek
Filename : draft-ietf-avt-mwpp-midi-rtp-05.txt
Pages : 94
Date : 2002-9-23

The MIDI Wire Protocol Packetization (MWPP) is a general-purpose RTP packetization for the MIDI command language. MWPP is suitable for use in both interactive applications (such as the remote operation of musical instruments) and content-delivery applications (such as MIDI file streaming). MWPP is suitable for use over unicast and multicast UDP, and defines tools that support the graceful recovery from packet loss. MWPP may also be used over reliable transport such as TCP. The SDP parameters defined for MWPP support the

customization of stream behavior (including the MIDI rendering method) during session setup. MWPP is compatible with the MPEG-4 generic RTP payload format, to support the MPEG 4 Audio object types for General MIDI, DLS2, and Structured Audio.

Title : Basic Socket Interface Extensions for IPv6
Author(s) : R. Gilligan, S. Thomson, J. Bound,
J. McCann, W. Stevens
Filename : draft-ietf-ipngwg-rfc2553bis-07.txt
Pages : 31
Date : 2002-9-19

The de facto standard application program interface (API) for TCP/IP applications is the 'sockets' interface. Although this API was developed for Unix in the early 1980s it has also been implemented on a wide variety of non-Unix systems. TCP/IP applications written using the sockets API have in the past enjoyed a high degree of portability and we would like the same portability with IPv6 applications. But changes are required to the sockets API to support IPv6 and this memo describes these changes. These include a new socket address structure to carry IPv6 addresses, new address conversion functions, and some new socket options. These extensions are designed to provide access to the basic IPv6 features required by TCP and UDP applications, including multicasting, while introducing a minimum of change into the system and providing complete compatibility for existing IPv4 applications. Additional extensions for advanced IPv6 features (raw sockets and access to the IPv6 extension headers) are defined in another document [4].

Title : MIPv6 Care of Address Option
Author(s) : A. O'Neill
Filename : draft-oneill-mipv6-cao-00.txt
Pages : 17
Date : 2002-9-19

IPv6 and MIPv6 has constrained the HoA to being used within forward and reverse tunnels via the HA. In the unicast case, the MN can then activate Route Optimisation to bypass the HA in both directions by securely installing a Binding Cache Entry into the CN. The MN then sends from the CCoA source address to the CN directly into the foreign multicast system, and includes the Home Address Option (HAO) so that the changing CCoA is masked from the transport layer. This draft defines the Care of Address Option, which carries the current CCoA of the MN. The CAO can be included in a Hop By Hop Header or Destination header and used instead of the packet source address for unicast ingress filtering and multicast RPF purposes. This enables a MN to potentially use the HoA as a source address on the foreign network, and to inform the CNs of the evolving MN location.

Title : A Quality-of-Service Resource Allocation Client for CASP
Author(s) : H. Schulzrinne et al.
Filename : draft-schulzrinne-nsis-casp-qos-00.txt

Pages : 12
Date : 2002-9-16

Signaling resource reservations is one of the possible applications of the Cross-Application Signaling Protocol (CASP). This document describes a client protocol that supports per-flow resource reservation for unicast and source-specific multicast flows, in both in-band and out-of-band modes, in sender- and receiver-directed operation.

Title : Aggregated Multicast: A Scheme to Reduce Multicast States
Author(s) : J. Cui et al.
Filename : draft-cui-multicast-aggregation-01.txt
Pages : 13
Date : 2002-9-9

In this document, we present a novel scheme, called aggregated multicast, to reduce multicast states ([FeiGI01] and [FeiNGC01]). The key idea is that multiple groups are forced to share one distribution tree, which we call aggregated tree. In our scheme, core routers need to keep states only per aggregated tree instead of per group. This can significantly reduce the total number of trees in the network and thus reduce forwarding states. We investigate the implementation issues of aggregated multicast in different network scenarios. We also discuss the effects of aggregated multicast on some important issues, such as QoS multicast provisioning, mobility support and fault tolerance. The scope of this paper is not to propose a detailed protocol, but present the idea of aggregated multicast at high level and show its merits.

Title : IPv6 Addressing Architecture Support for mobile ad hoc networks
Author(s) : G. Chelius, E. Fleury
Filename : draft-chelius-adhoc-ipv6-00.txt
Pages : 10
Date : 2002-9-5

The concept of node identifier, in practical terms an IP address, is crucial in ad hoc networks. Its use allows the setup of IP routing for ad hoc connectivity and the identification of several wireless devices as part of a unique ad hoc node. In this document, a new addressable object is defined: the ad hoc connector. It virtualizes several ad hoc network interfaces into a single addressable object. To locally address ad hoc connectors, a third IPv6 local-use unicast address (adhoc-local address) and the correlated use of the subnet multicast scope are defined.

Title : Multicast Announce and Control Protocol (MACP)
Author(s) : J. Helal
Filename : draft-helal-macp-00.txt
Pages : 49
Date : 2002-9-3

This memo describes the message and procedure related to the Multicast Announce and

Control Protocol (MACP). This protocol is considered as one 'building blocks' of a reliable multicast transport framework. The Multicast Announce and Control Protocol (MACP) organizes the process by which a multicast sender node (Msender) manages transmissions to dynamic groups of receivers (Mreceivers) in the 'one-to-many' model. The prime objective of MACP is to work in conjunction with various data transport protocols in order to meet various network requirements. One other main objective is to provide a unified announce transport mechanism for both bulk data transfer and streamed data.

Title : Linklocal Multicast Name Resolution (LLMNR)
Author(s) : L. Esibov, B. Aboba, D. Thaler
Filename : draft-ietf-dnsext-mdns-12.txt
Pages : 20
Date : 2002-8-27

Today, with the rise of home networking, there are an increasing number of ad-hoc networks operating without a DNS server. In order to allow name resolution in such environments, Link-Local Multicast Name Resolution (LLMNR) is proposed. LLMNR supports all current and future DNS formats, types and classes, while operating on a separate port from DNS, and with a distinct resolver cache.

Title : Explicit multicast reachability test
Author(s) : J. Lee
Filename : draft-lee-xcast-reachability-00.txt
Pages : 11
Date : 13-Aug-02

It can be important to know which node has Explicit multicast receivability and routability before sending a large amount of data traffic in Explicit multicast. This document provides how to test the receivability and routability, in short, the reachability of Explicit multicast packets to a particular node.

Title : Explicit Multicast Tunneling
Author(s) : J. Lee
Filename : draft-lee-xcast-tunneling-01.txt
Pages : 21
Date : 09-Aug-02

Explicit multicast(Xcast)[1] is a new kind of Internet multicast, and encodes the list of destinations within its packet. This document specifies tunneling scheme of Xcast packets. Since a single Xcast tunnel has multiple egress-nodes, the original Xcast packet can be encapsulated either within a Xcast packet or within a unicast packet. When tunneled by Xcast-in-Xcast encapsulation, the bitmap of the original Xcast packet is overwritten by the bitmap of the tunnel Xcast packet at tunnel egress-nodes, in order to control the active destination set.

Title : Host Extensions to Protocol Independent Multicast
Author(s) : K. Patel, R. Perlman
Filename : draft-keyur-pim-host-extensions-00.txt
Pages :
Date : 06-Aug-02

This document defines host extensions to Protocol Independent Multicast - PIM protocol. These host extensions allows endnodes to join/leave any multicast (S/*,G) groups. This helps in easing SSM-style multicast deployment that does not have to depend on IGMP (v1/v2/v3), in either endnodes or the routers.

Title : L2TP Multicast Extension
Author(s) : G. Bourdon
Filename : draft-ietf-l2tpext-mcast-02.txt
Pages : 16
Date : 01-Aug-02

The Layer Two Tunneling Protocol (L2TP) [RFC2661] provides a standard method for tunneling PPP [RFC1661] packets. This document describes an extension to L2TP, in order to have an efficient use of L2TP tunnels within the context of deploying multicast services whose data will have to be conveyed by such tunnels.

Title : Mobility Management and IP Multicast
Author(s) : A. O'Neill
Filename : draft-oneill-mip-multicast-00.txt
Pages : 20
Date : 25-Jul-02

Mobile IP provides a mobile node, that visits a foreign subnet, the ability to continue to use an address from its home subnet (the home address) as a source address. This is achieved through the allocation of a Care of Address on the foreign subnet that is used as the end-point of a redirection tunnel from a home agent on the home subnet. Mobile IP in RFC 3220 states that when the mobile node originates multicast traffic intended for the foreign multicast system, it can only do so by first obtaining an IP address from the foreign subnet (a Collocated Care of Address) and then using this address as the multicast source address. This is to ensure that the source address will pass multicast routing reverse path forwarding checks. This foreign multicast model is however extremely restrictive, and still very problematic to multicast routing and applications when the mobile node regularly changes foreign subnets, as is common in wireless systems. This is because the source address continues to evolve which must be tracked by source specific multicast application and routing signalling. Using the home multicast system, again described above, is also non-optimal because the mobile node receiver is then serviced by packets that must be tunnelled from its home agent which, removes any multicast routing benefits (ie network based tree building). This draft therefore describes modifications to

the foreign multicast interface between mobile IP and multicast routing that enable the mobile node to use its persistent home address as a multicast source address.

Title : The Bordercast Resolution Protocol (BRP) for Ad Hoc Networks
Author(s) : Z. Haas, M. Pearlman, P. Samar
Filename : draft-ietf-manet-zone-brp-02.txt
Pages : 13
Date : 05-Jul-02

The Bordercast Resolution Protocol (BRP) provides the bordercasting packet delivery service used to support network querying applications. The BRP uses a map of an extended routing zone, provided by the local proactive Intrazone Routing Protocol (IARP), to construct bordercast (multicast) trees, along which query packets are directed. Within the context of the hybrid ZRP, the BRP is used to guide the route requests of the global reactive Interzone Routing Protocol (IERP). The BRP employs special query control mechanisms to steer route requests away from areas of the network that have already been covered by the query. The combination of multicasting and zone based query control makes bordercasting an efficient and tunable service that is more suitable than flood searching for network probing applications like route discovery.

Title : Securing Group Management in IPv6 with Cryptographically Generated Addresses
Author(s) : C. Castelluccia, G. Montenegro
Filename : draft-irtf-gsec-sgmv6-01.txt
Pages : 32
Date : 05-Jul-02

Currently, group membership management in IP Multicast and Anycast can be abused in order to launch denial-of-service (DoS) attacks. The root of the problem is that routers cannot determine if a given host is authorized to join a group (sometimes referred to as the 'Proof-of-Membership Problem' [ECUMN00]). We propose a solution for IPv6 based on Group Cryptographically Generated Addresses (G-CGA). These addresses have characteristics of statistical uniqueness and cryptographic verifiability that lend themselves to severely limiting certain classes of DoS attacks. Our scheme is fully distributed and does not require any trusted third party or pre-established security association between the routers and the hosts. This is not only a huge gain in terms of scalability, reliability and overhead, but also in terms of privacy.

Title : IP Multicast in Differentiated Services Networks
Author(s) : R. Bless, K. Wehrle
Filename : draft-bless-diffserv-multicast-04.txt
Pages : 34
Date : 05-Jul-02

This document presents some of the problems which will arise when IP Multicast is used in DiffServ networks without taking special precautions into account for supplying multicast

services. Although the basic DS forwarding mechanisms also work with IP Multicast, some facts have to be considered which are related to the provisioning of multicast resources. The presented problems mainly lead to situations in which other service users are affected adversely in their experienced quality. In order to retain the benefits of the DiffServ approach, a quite simple and scalable solution for those problems is required, not resulting in additional complexity or costs related to forwarding mechanisms in a DiffServ domain

---

Title : Link Scoped IPv6 Multicast Addresses
Author(s) : J. Park, M. Shin, Y. Kim
Filename : draft-ietf-ipv6-link-scoped-mcast-01.txt
Pages : 6
Date : 05-Jul-02

This document specifies an extension to the multicast addressing architecture of the IPv6 protocol. The extension allows for the use of interface-ID to allocate multicast addresses. When the link-local unicast address is configured at each interface of host, interface ID is uniquely determined. By delegating multicast addresses at the same time as interface ID, each host can identify their multicast addresses automatically at Layer 1 without running an intra- or inter-domain allocation protocol in the serverless environments.

Title : RTCP Extensions for Single-Source Multicast Sessions with Unicase Feedback
Author(s) : J. Chesterfield et al.
Filename : draft-ietf-avt-rtcpssm-01.txt
Pages : 28
Date : 05-Jul-02

This document specifies a modification to the Real-time Transport Control Protocol (RTCP) to use unicast feedback. The proposed extension is useful for single source multicast sessions such as Source Specific Multicast (SSM) communication where the traditional model of many-to-many group communication is either not possible or not preferred. In addition, it can be applied to any group that might benefit from a sender controlled summarised reporting mechanism.

Back

IP Multicast

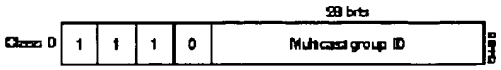
The Internet Engineering Task Force (IETF) has developed standards that address the parameters that are required to support multicast communications:

- *Addressing*--The IP address space is divided into four sections: Class A, Class B, Class C, and Class D. Class A, B, and C addresses are used for unicast traffic. Class D addresses are reserved for multicast traffic and are allocated dynamically.
- *Dynamic registration*--RFC 1112 defines the Internet Group Management Protocol (IGMP). IGMP specifies how the host should inform the network that it is a member of a particular multicast group.
- *Multicast routing*--There are several standards for routing IP multicast traffic:
    - Distance Vector Multicast Routing Protocol (DVMRP) as described in RFC 1075.
    - Multicast Open Shortest Path First (MOSPF), which is an extension to Open Shortest Path First (OSPF) that allows it to support IP multicast, as defined in RFC 1584.
    - Protocol Independent Multicast (PIM), which is a multicast protocol that can be used with all unicast IP routing protocols, as defined in the two Internet standards-track drafts entitled *Protocol Independent Multicast (PIM): Motivation and Architecture* and *Protocol Independent Multicast (PIM): Protocol Specification*.

IP Multicast Group Addressing

Figure 13-12 shows the format of a Class D IP multicast address.

Figure 13-12: Class D Address Format



Unlike class A, B, and C IP addresses, the last 28 bits of a Class D address have no structure. The multicast group address is the combination of the high-order 4 bits of 1110 and themulticast group ID. These are typically written as dotted-decimal numbers and are in the range 224.0.0.0 through 239.255.255.255. Note that the high-order bits are 1110. If the bits in the first octet are 0, this yields the 224 portion of the address.

The set of hosts that responds to a particular IP multicast address is called a *host group*. A host group can span multiple networks. Membership in a host group is dynamic--hosts can join and leave host groups. For a discussion of IP multicast registration, see "Internet Group Management Protocol," later in this chapter.

Some multicast group addresses are assigned as well-known addresses by the Internet Assigned Numbers Authority (IANA). These multicast group addresses are called*permanent host groups*

and are similar in concept to the well-known TCP and UDP port numbers. Address 224.0.0.1 means "all systems on this subnet," and 224.0.0.2 means "all routers on this subnet."

Table 13-6 list the multicast address of some permanent host groups.

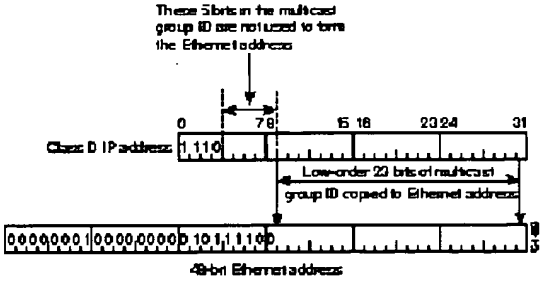
Table 13-6: Example of Multicast Addresses for Permanent Host Groups.

Permanent Host Group	Multicast Address
Network Time Protocol	224.0.1.1
RIP-2	224.0.0.9
Silicon Graphics Dogfight applicatior	224.0.1.2

The IANA owns a block of Ethernet addresses that in hexadecimal is 00:00:5e. This is the high-order 24 bits of the Ethernet address, meaning that this block includes addresses in the range 00:00:5e:00:00:00 to 00:00:5e:ff:ff:ff. The IANA allocates half of this block for multicast addresses. Given that the first byte of any Ethernet address must be 01 to specify a multicast address, the Ethernet addresses corresponding to IP multicasting are in the range 01:00:5e:00:00:00 through 01:00:5e:7f:ff:ff.

This allocation allows for 23 bits in the Ethernet address to correspond to the IP multicast group ID. The mapping places the low-order 23 bits of the multicast group ID into these 23 bits of the Ethernet address, as shown in Figure 13-13. Because the upper 5 bits of the multicast address are ignored in this mapping, the resulting address is not unique. Thirty-two differentmulticast group IDs map to each Ethernet address.

Figure 13-13: Multicast Address Mapping



Because the mapping is not unique and because the interface card might receive multicast frames in which the host is really not interested, the device driver or IP modules must perform filtering.

Multicasting on a single physical network is simple. The sending process specifies a destination IP address that is a multicast address, and the device driver converts this to the corresponding Ethernet address and sends it. The receiving processes must notify their IP layers that they want to receive datagrams destined for a given multicast address and the device driver must somehow enable reception of these multicast frames. This process is handled by joining a multicast group.

When a multicast datagram is received by a host, it must deliver a copy to all the processes that

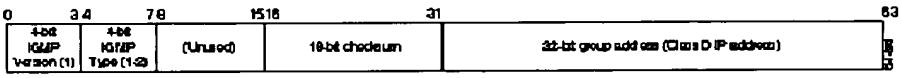
belong to that group. This is different from UDP where a single process receives an incoming unicast UDP datagram. With multicast, multiple processes on a given host can belong to the same multicast group.

Complications arise when multicasting is extended beyond a single physical network and multicast packets pass through routers. A protocol is needed for routers to know if any hosts on a given physical network belong to a given multicast group. This function is handled by IGMP.

Internet Group Management Protocol

The Internet Group Management Protocol (IGMP) is part of the IP layer and uses IP datagrams (consisting of a 20-byte IP header and an 8-byte IGRP message) to transmit information about multicast groups. IGMP messages are specified in the IP datagram with a protocol value of 2. Figure 13-14 shows the format of the 8-byte IGMP message.

Figure 13-14: IGMP Message Format



The value of the *version* field is 1. The value of the *type* field is 1 for a query sent by a multicast router and 2 for a report sent by a host. The value of the *checksum* field is calculated in the same way as the ICMP checksum. The group address is a class D IP address. In a query, the group address is set to 0, and in a report, it contains the group address being reported.

The concept of a process joining a multicast group on a given host interface is fundamental to multicasting. Membership in a multicast group on a given interface is dynamic (that is, it changes over time as processes join and leave the group). This means that end users can dynamically join multicast groups based on the applications that they execute.

Multicast routers use IGMP messages to keep track of group membership on each of the networks that are physically attached to the router. The following rules apply:

- A host sends an IGMP report when the first process joins a group. The report is sent out the same interface on which the process joined the group. Note that if other processes on the same host join the same group, the host does *not* send another report.
- A host does not send a report when processes leave a group, even when the last process leaves a group. The host knows that there are no members in a given group, so when it receives the next query, it doesn't report the group.
- A multicast router sends an IGMP query at regular intervals to see if any hosts still have processes belonging to any groups. The router sends a query out each interface. The group address in the query is 0 because the router expects one response from a host for every group that contains one or more members on a host.
- A host responds to an IGMP query by sending one IGMP report for each group that still

contains at least one process.

Using queries and reports, a multicast router keeps a table of its interfaces that have one or more hosts in a multicast group. When the router receives a multicast datagram to forward, it forwards the datagram (using the corresponding multicast OSI Layer 2 address) on only those interfaces that still have hosts with processes belonging to that group.

The Time to Live (TTL) field in the IP header of reports and queries is set to 1. A multicast datagram with a TTL of 0 is restricted to the same host. By default, a multicast datagram with a TTL of 1 is restricted to the same subnet. Higher TTL field values can be forwarded by the router. By increasing the TTL, an application can perform an expanding ring search for a particular server. The first multicast datagram is sent with a TTL of 1. If no response is received, a TTL of 2 is tried, then 3, and so on. In this way, the application locates the server that is closest in terms of hops.

The special range of addresses 224.0.0.0 through 224.0.0.255 is intended for applications that never need to multicast further than one hop. A multicast router should never forward a datagram with one of these addresses as the destination, regardless of the TTL.

Multicast Routing Protocols

A critical issue for delivering multicast traffic in a routed network is the choice of multicast routing protocol. Three different multicast routing protocols have been defined for this purpose:

- Distance Vector Multicast Routing Protocol
- Multicast OSPF
- Protocol Independent Multicast

The goal in each protocol is to establish paths in the network so that multicast traffic can effectively reach all group members.

Distance Vector Multicast Routing Protocol

Distance Vector Multicast Routing Protocol (DVMRP) uses a technique known as *reverse path forwarding*. When a router receives a packet, it floods the packet out all paths except the path that leads back to the packet's source. Reverse path forwarding allows a data stream to reach all LANs (possibly multiple times). If a router is attached to a set of LANs that does not want to receive a particular multicast group, the router sends a "prune" message up the distribution tree to prevent subsequent packets from traveling where there are no members.

New receivers are handled by using grafts. Consequently, only one round-trip time (RTT) from the new receiver to the nearest active branch of the tree is required for the new receiver to start getting traffic.

To determine which interface leads back to the source of the data stream, DVMRP implements its own unicast routing protocol. This unicast routing protocol is similar to RIP and is based on hop counts. As a result, the path that the multicast traffic follows might not be the same as the path that the unicast traffic follows.

REAL-TIME OPTIMAL MULTICAST ROUTING

Author: ***Dechanuchit Katanyutaveetip***

by Ozgur Savas

1. Introduction

Multicasting is a communication technique that allows a source to transmit multiple copies of a packet to a set of receivers. Multicasting can be achieved independent from location information. Multicasting was not clearly understood and achieved between remote locations until the multicast IP address space was defined and separated from the actual IP space as class D addresses. Multicasting on a local subnetwork does not require either the presence of a multicast router or the implementation of a multicast routing algorithm, on a shared media such as Ethernet. If the transmission is in a local subnetwork, sender does not really need to be a group member, simply sends a multicast data packet that is received by any member hosts connected to the same medium.

For multicasts beyond the scope of local subnetwork, the subnet must have a multicast capable router attached, which itself is (probably virtually) attached to another multicast capable router and so on. The collection of these virtually connected routers forms the Internet's multicast backbone also known as Mbone.

With Mbone, a single packet can have multiple destinations and isn't split up until the last possible moment. This means that it can pass through several routers before it needs to be divided to reach its final destinations. This leads much more efficient transmission and also ensures that packets reach multiple destinations roughly at the same time.

Multicast routing protocols use IGMP to monitor group membership position on its directly attached links, so that if any multicast data arrives it knows which of its links to send a copy of a packet. Actually, Mbone does not use the same routing protocol for all the multicast capable routers attached to its backbone, instead there are many protocols used to interconnect different regions. These protocols can be classified into two main groups according to tree structures they use, as *Source-Based Tree* and *Shared-Tree Algorithms*. The main protocols that use these algorithm structures are defined in the following section.

The main concern of this article is to present a new multicast algorithm by modifying the path selection technique that is used by well-known shared-tree algorithm; *Core-Based Tree (CBT).*

In the following section, Multicast Routing Protocols depending on the structures mentioned are explained deeper, in the third section real-time multicast network model is explained and the modifications on CBT are given, in the fourth section simulation results are summarized to prove the positive affects of modifications and the paper concluded with the fifth section.

2. Multicast Routing Protocols

2.1. Source Based Trees (SBT)

Data packets addressed to a multicast group may be routed on a tree that is specific to the particular sender and group. In this approach all the senders create a unique shortest path tree rooted by themselves.

Source-based multicast trees are either built by a distance-vector style algorithm, which may be implemented separately from the unicast routing algorithm (as in DVMRP), or the multicast tree may be built using the information present in the underlying unicast routing table (as in PIM-DM). The other algorithm used for building source-based trees is the link state algorithm as Multicast OSPF – MOSPF.

2.1.1. Distance-Vector Multicast Routing Protocol (DVMRP)

When a multicast router receives a multicast packet, if the packet arrives on the interface used to reach the source of the packet, the packet is forwarded over all outgoing interfaces, except leaf subnets with no members attached. If the data packet does not arrive over the link that would be used to reach the source, the packet is discarded.

When a packet arrives to a leaf router if that router has no membership on any of its directly attached networks, router sends a *prune* message one hop back towards the source. The receiving router then checks its leaf subnets for group membership and checks whether it has received any prune messages from all directly attached routers backwards on the way from the source. If so the router itself sends a prune message over the interfaces. The prune messages are kept in routers for a lifetime, this information should be refreshed by neighbors, or it will be removed.

2.1.2. Link-State Multicast Routing Protocol (MOSPF)

All the routers in the subnetwork collect reachability information from their directly attached neighbors and flood out this information periodically to the network, each router then has a complete fresh information about the topology of the network and the group memberships. On receiving a multicast packet, each router uses its membership and topology information to create a shortest-path tree rooted at the sender's subnetwork. Sender router puts the packets on its interfaces to pass it away to neighbors according to that tree information, all the members on the way repeats this action and the way to calculate a source based tree over Mbone happens.

2.2. Shared Trees (Core-Based Tree - CBT)

As it is easily seen the source-based tree algorithms are not efficient on large amount of members and groups. This makes them less scalable. A shared tree architecture offers an improvement in scalability over source tree architectures by a factor of the number of active sources. Despite, all the senders have to create a separate tree itself for each tree, in source tree algorithms there is only one shared tree for each group. The tree is a shortest path tree rooted at one or more predefined nodes in the network called *core nodes.Core Based Tree (CBT)* architecture makes the multicast topology more scalable, robust, simple and interoperable.

For each network multicast group, a router is selected as a Designated Router (DR) which is the core of the multicast group. The multicast tree is being created with SPF algorithm by the core including all the members. If any multicast capable router wants to join to the multicast group it sends a JOIN_REQUEST to the DR. This join message must be acknowledged with JOIN_ACK by any member of the multicast tree or the core's itself. Once the acknowledge reaches the router that originated the join message the new receiver can receive the traffic sent to the group. If an on tree router wants to leave the group it sends a QUIT_NOTIFICATION to its parent router. Notification message is sent upstream also in the case of lack of response to an ECHO_REQUEST. There is also FLUSH_TREE message which is sent downstream. If any router receiving that message has any group members attached, it restarts the joining process to the group's core. ECHO_REQUEST and ECHO_REPLY packets are used to maintain the multicast tree. ECHO_REQUEST does not contain any group information, the ECHO_REPLY does but only periodically. To maintain consistent information between parent and child, the parent periodically reports ECHO_REPLYs.

If we compare source based tree algorithms with shared tree, it can be said that source based tree mechanisms scales worse than shared tree. Table sizes in source based trees are larger than core based by a factor of number of sources. Also the overhead in the network is larger than CBT. DVMRP periodically sends prune messages and MOSPF periodically floods link states. On the other hand, the disadvantage of CBT is the single point of failure structure. Also delay is larger in CBT because the shared tree may not be the shortest path for all senders in multicast tree.

3. Real-time Multicast Networks Model

In real time communication, the primary criteria for a network packet transmission is the real time constraint. All the messages from a sender should be received by destinations within a specified time delay. In real time communications there is a channel establishment state between communicators, which should be satisfied within the sufficient network resources, time delay and bandwidth.

There are two main algorithms used to construct a shortest path tree, which are Dijkstra and Bellman-Ford. Dijkstra minimizes the cost of the route and Bellman-Ford minimizes the number of links in the tree. However, we need the construction of a real-time communication of a tree that guarantees the success of communication within a specified time constraint and also the careful usage of network resources. In order to achieve this aim the author of this paper demonstrates a new shortest path algorithm, *weighted Dijkstra's shortest path algorithm.* This algorithm depends on the well known Dijkstra's algorithm but it uses a weighted metric instead of the default cost metric of the former one. Since building a real-time multicast tree requires optimizing of the network cost while meeting the delay constraint, both the cost and delay factors can be considered at once and the optimal path can be defined as, *(Cw * Path Cost) + (Dw * Path Delay)* where Cw is the Cost Weight and the Dw is the Delay Weight.

In the former Dijkstra's algorithm the main metric was the *cost* which represents *hop count* and the *distance.* In the new algorithm the cost metric is replaced with the total of the cost and delay metrics.

Weighted Dijkstra's Shortest Path Tree Algorithm

G = set of nodes in a network

S = set of nodes in the weighted shortest path tree

s = source node

$Cost_{x,y}$ = path cost from node x to node y

$Delay_{x,y}$ = path delay from node x to node y

$CD_{x,y} = (Cw * Cost_{x,y}) + (Dw * Delay_{x,y})$

Procedure weighted_Dijkstra ;

1. S ¬ {s}
2. Cost[n] = $Cost_{x,y}$
3. Delay[n] = $Delay_{x,y}$
4. while S <> G
5. Find w not an element of S such that $CD_{w,s} = \min CD_{j,s}$
6. S ¬ S È {w}
7. For all n not an element of S
8. if ($CD_{n,w} + CD_{w,s} < CD_{n,s}$)
9. Cost[n]= $Cost_{n,w}$ + Cost[w] ; Delay[n]= $Delay_{n,w}$ + Delay[w]
10. For each destination node if the path delay is greater than delay constraint T, replace the path by the shortest path based on delay only.
11. Remove all branches leading to non-destination nodes.

CBT Modifications

In the modified CBT, if a node out of the tree wants to send a multicast packet as a source to the members of tree, first all the shortest paths for all the nodes on tree are constructed then the shortest of these paths are chosen as the *shortest of the shortest paths.* Applying that path the weighted Dijkstra gives the shortest distances to all multicast nodes on tree. This procedure guarantees that the transmission via all the paths from a source to any node is achieved in a delay constrained.

4. Simulation Results Summary

There are four metrics considered to see the performance impact of new demonstrated algorithm and modification on CBT. The metrics and the results that were observed are ;

Average end-to-end delay: The two discrete simulations with number of group members 50 and 100 show that, new constructed protocol yields better cost and delay performance than those of the original CBT for both group sizes.

Network resource usage: The simulations with number of group members 50 and 100 show that, the network resource usage of new protocol is less than that of the CBT. It is experimented by varying different end-to-end delay time constraint values and seen that modified protocol uses fewer hop counts than its counterpart to achieve the transmission between nodes. Furthermore, as multicast group size increases modified protocol produces less costly networks than the original CBT.

Traffic concentration: The simulations for 50 and 100 multicast group members with an assumption of constant unit rate of traffic generation for each node show that, modified CBT achieves much more better link utilization about %60 better performance, and the maximum link load and the possibility of congestion errors are decreased.

Loss Rate: Since one of the most important criteria for creating the new algorithm was the real time delay constraint, it is seen as expected that the loss rate in modified CBT is less than the original one.

5. Conclusion

In this paper, a new approach to create a multicast routing tree within a specific delay constraint is presented. The new approach is a modification of well-known CBT protocol but the author added new capabilities to that process by taking care of real time applications and its requirements. The Dijkstra's shortest path algorithm that multicast network designers used to use for determining the shortest path is modified as weighted algorithm by replacing path selection criteria which is *cost* with a new metric as a sum of cost and *delay.* The simulation results showed that the new CBT protocol which uses weighted Dijkstra for path selection, obtains much more efficient, less congested, better utilized and more reliable results than its counterpart. Also with modified CBT came over the problem of optimal core node selection problem in former CBT by selecting the on-tree node which is the neighbor node –gateway of tree- of the new coming source node as core and distribution point. As a result, the path length from the new source to the branches of the tree -multicast group members- is minimized.